Conservative Visibility Preprocessing using Extended Projections

Frédo Durand, George Drettakis,
Joëlle Thollot and Claude Puech

iMAGIS-GRAVIR/IMAG-INRIA (Grenoble, France)
Laboratory for Computer Science – MIT (USA)

Special thanks

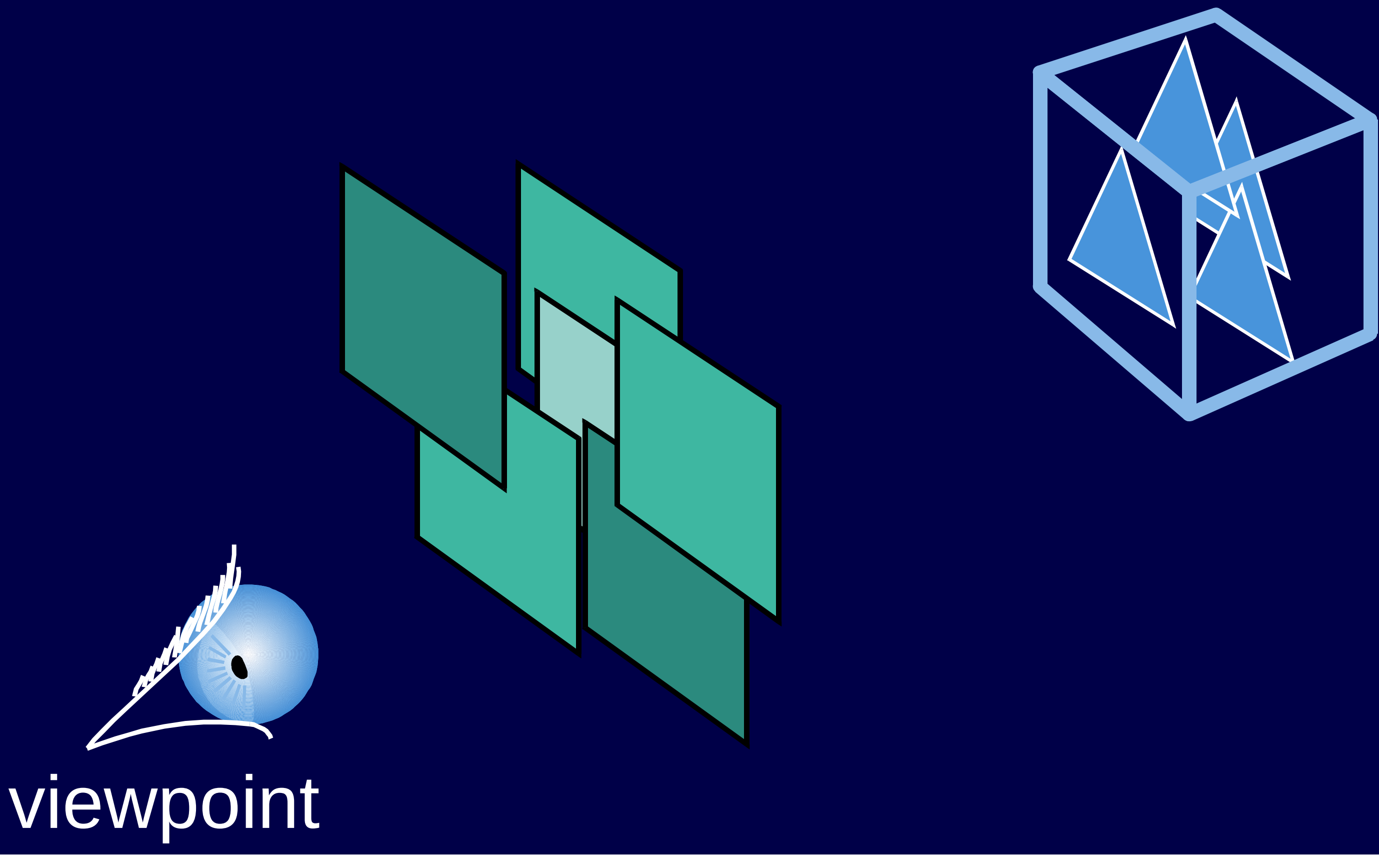
- Leo Guibas
- Mark de Berg

Introduction

- Walkthrough of large models
  - Simulators, games, CAD/CAM, urban planning
  - Millions of polygons
  - Not real-time with current graphics hardware
- Acceleration
  - Geometric Levels of Detail (LOD)
  - Image-based simplification (impostors)
  - View Frustum culling
  - Occlusion-culling

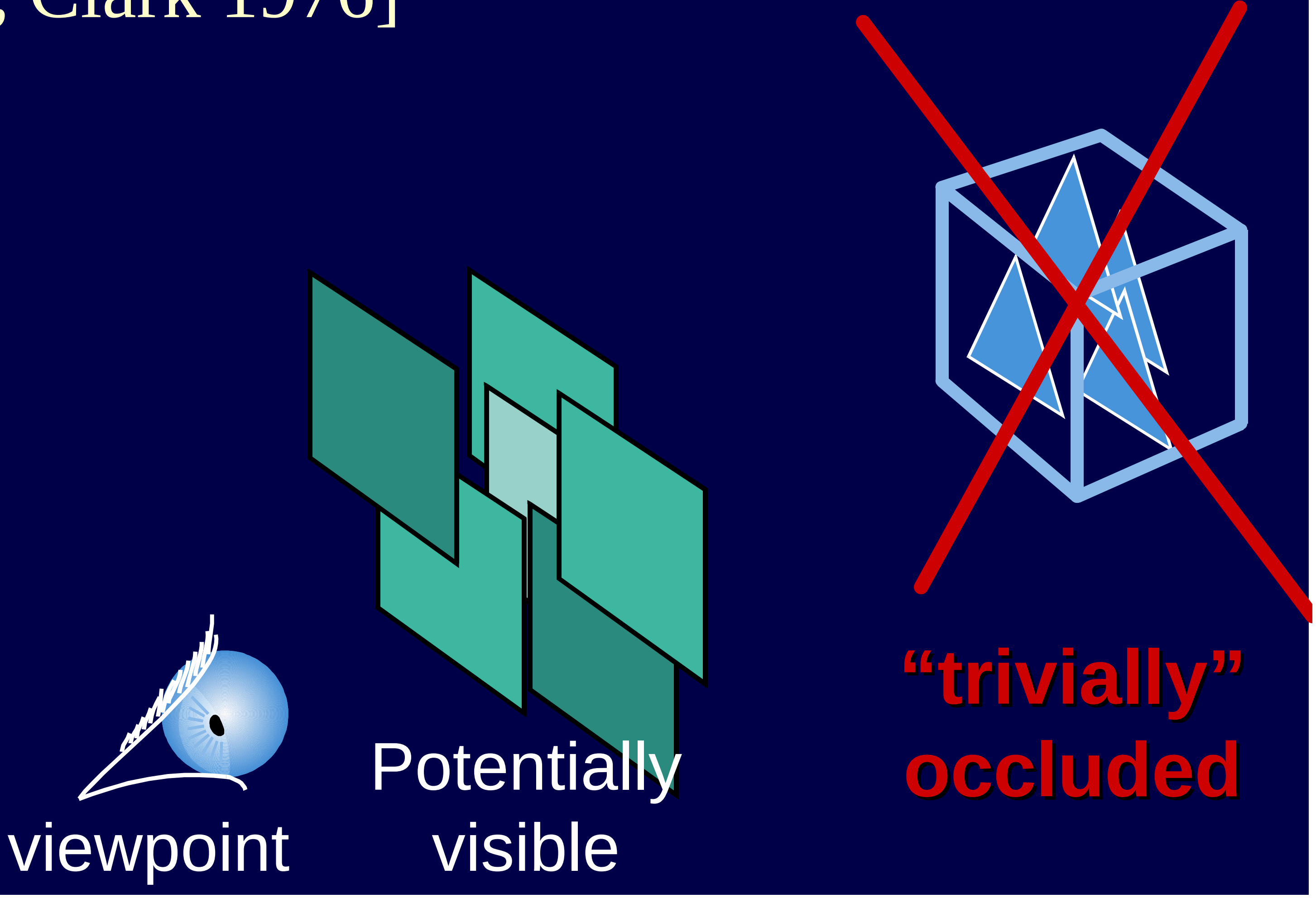
Occlusion culling - Principle

- Quickly reject hidden geometry
- [Jones 71, Clark 1976]



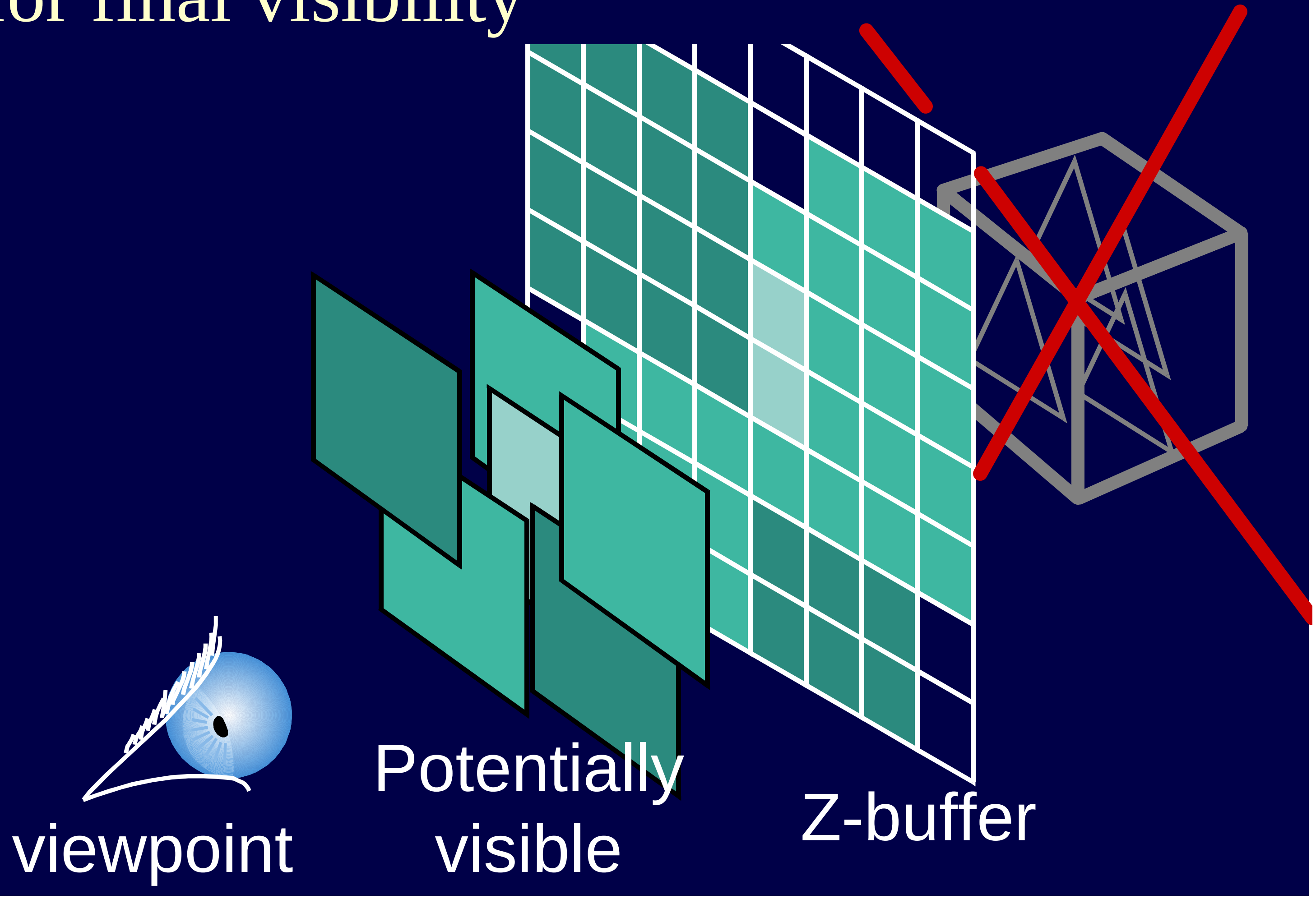
Occlusion culling - Principle

- Quickly reject hidden geometry
- [Jones 71, Clark 1976]



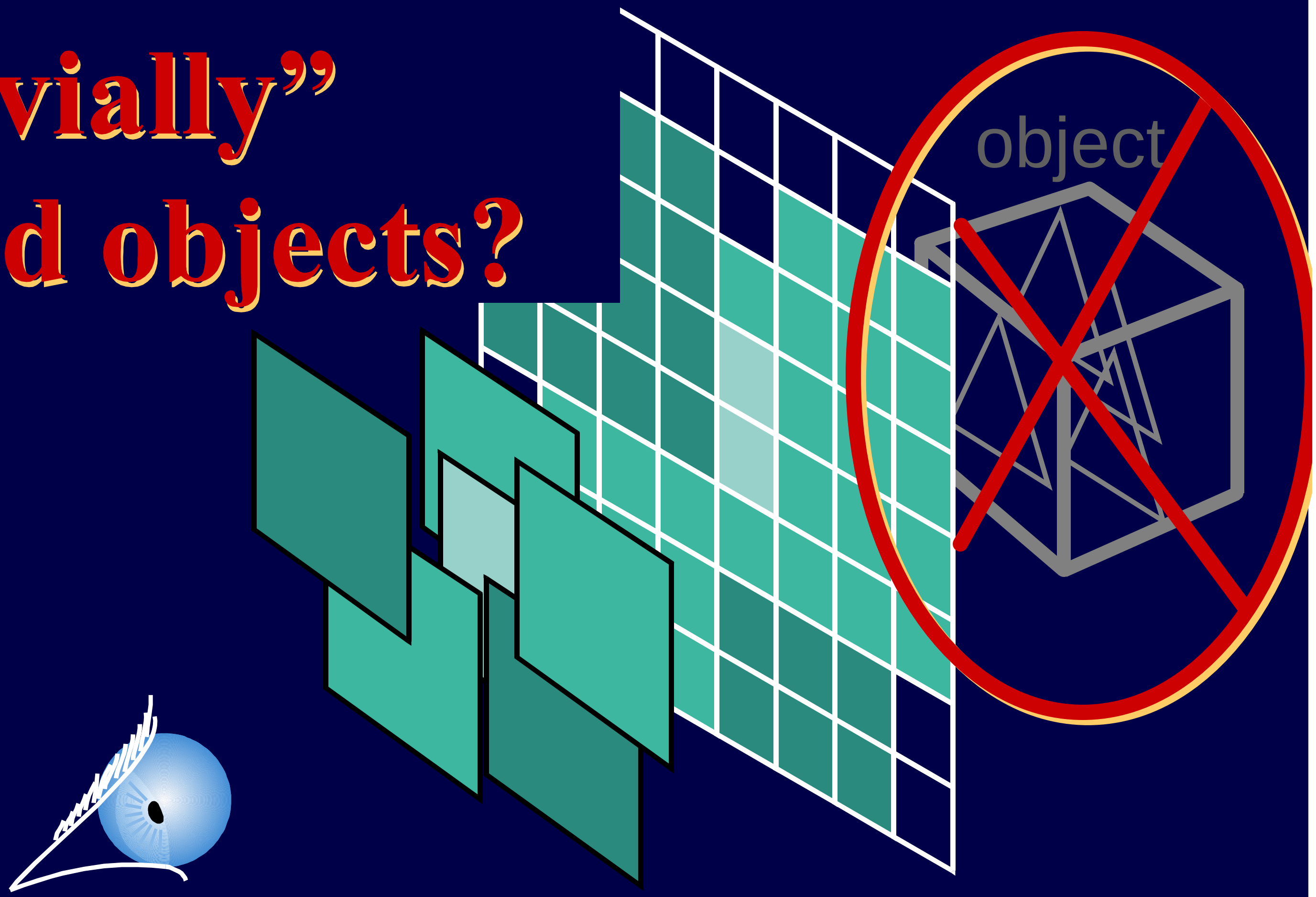
Occlusion culling - Principle

- Quickly reject hidden geometry
- Z-buffer for final visibility



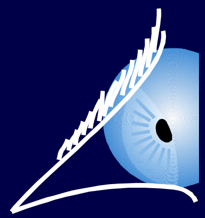
Occlusion culling - Problem

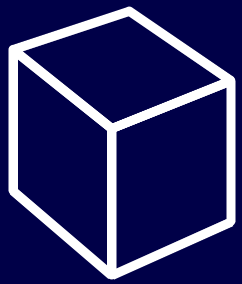
- **How can we detect the “trivially” occluded objects?**



Occlusion culling - Classification

- Online point-based / Preprocessing (cells)



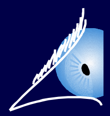


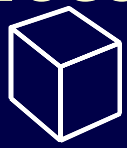
[Greene 93, Coorg 96, Zhang 97,
Luebke 95, etc.]

[Teller 91, Airey 91,
Cohen-Or 98, etc.]

Occlusion culling - Classification

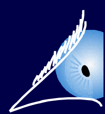
- Online point-based / Preprocessing (cells)

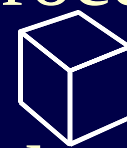




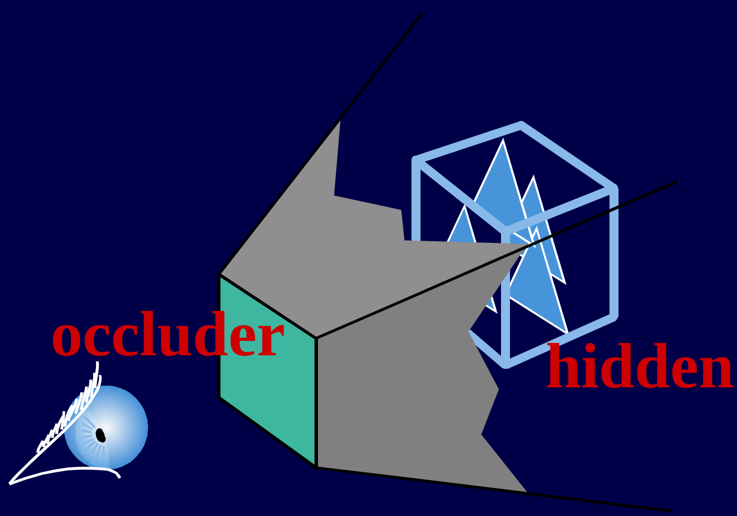
Occlusion culling - Classification

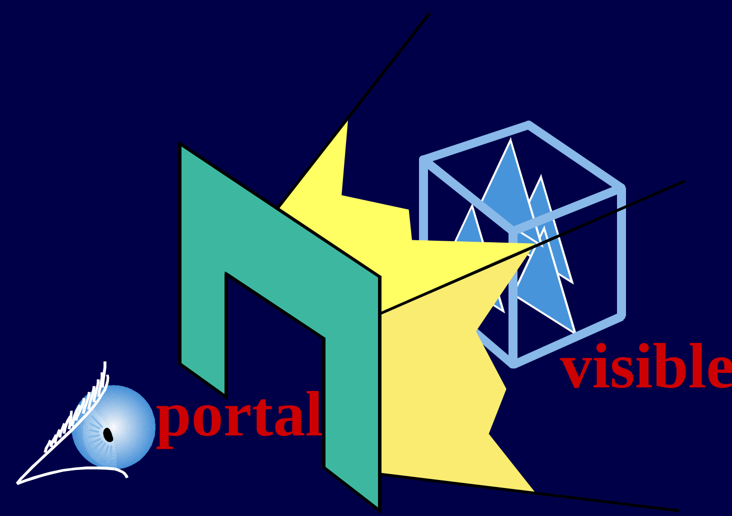
- Online point-based / Preprocessing (cells)





- Occluders / Portals



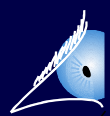


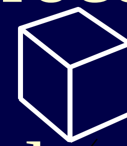
[Greene 93, Coorg 96, Zhang 97, Cohen-Or 98, etc.]

[Teller 91, Airey 91, Luebke 95, etc.]

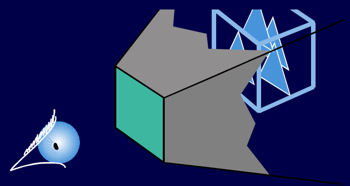
Occlusion culling - Classification

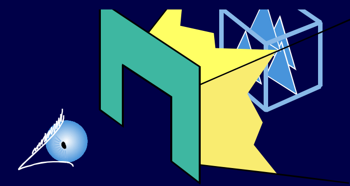
- Online point-based / Preprocessing (cells)





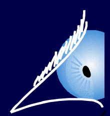
- Occluders / Portals

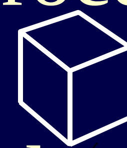




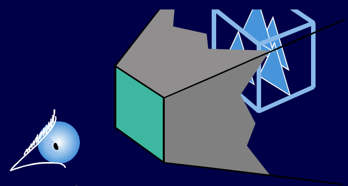
Occlusion culling - Classification

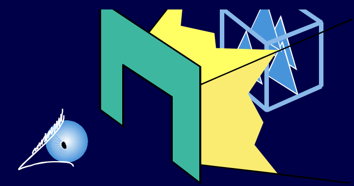
- Online point-based / Preprocessing (cells)



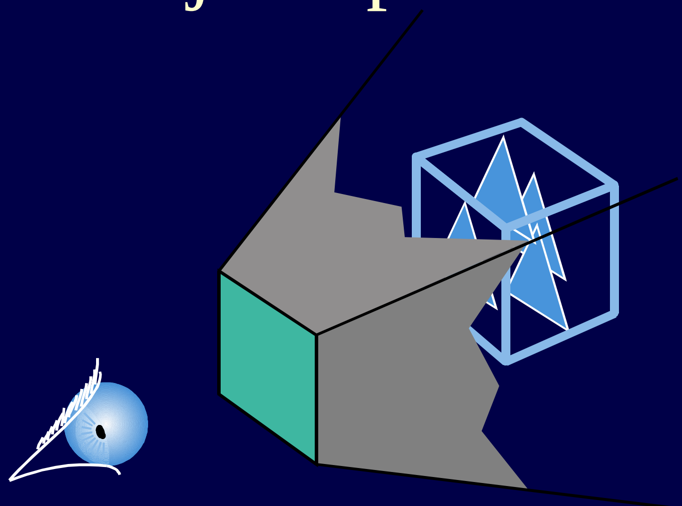


- Occluders / Portals

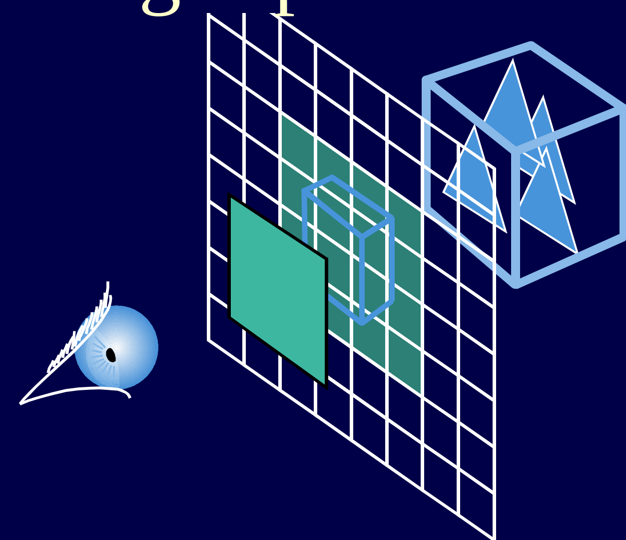




- Object space / Image space



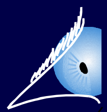
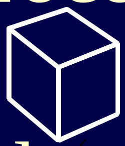
[Teller 91, Airey 91, Coorg 96, Hudson 97, Cohen-Or 98, etc.]



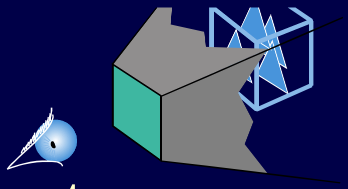
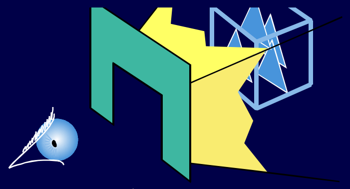
[Greene 93, Zhang 97, etc.]

Occlusion culling - Classification

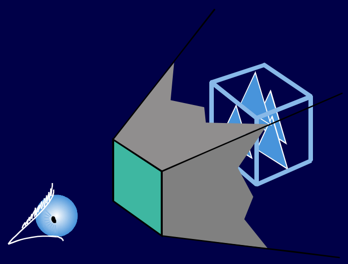
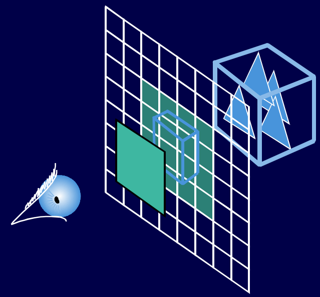
- Online point-based / Preprocessing (cells)

- Occluders / Portals

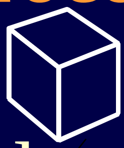



- Object space / Image space

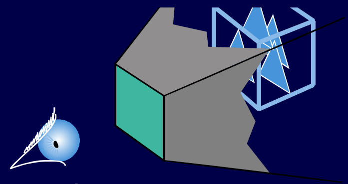



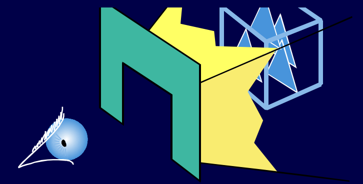
Occlusion culling - Classification

- Online point-based / Preprocessing (cells)

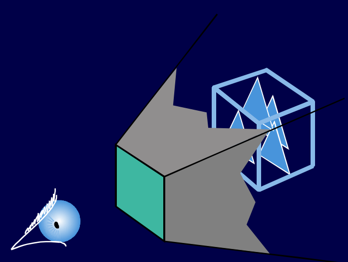


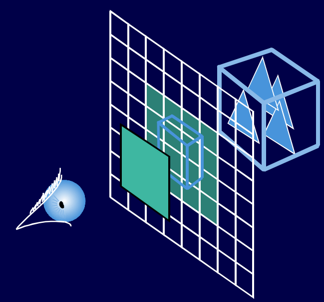
- Occluders / Portals





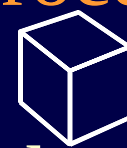
- Object space / Image space



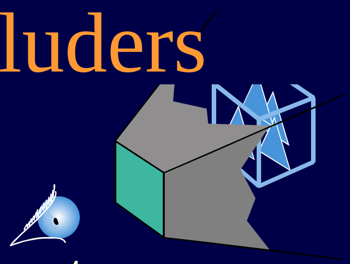


Occlusion culling - Classification

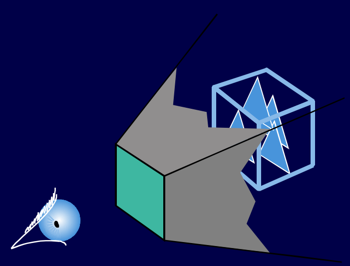
- Online point-based / Preprocessing (cells)

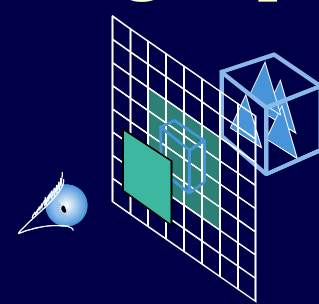


- Occluders / Portals



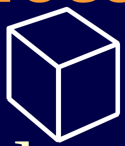
- Object space / Image space



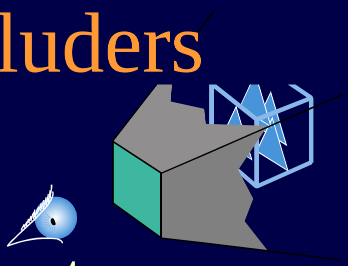


Occlusion culling - Classification

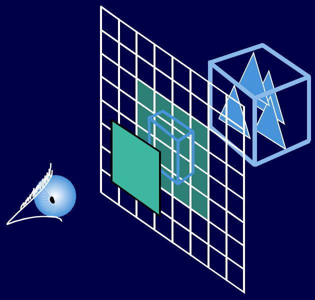
- Online point-based / Preprocessing (cells)



- Occluders / Portals



- Object space / Image space



Our approach

- Visibility preprocess
  - Objects invisible from a volumetric cell
- Conservative computation
  - Do not declare a visible object hidden
- Occluder fusion
  - Occlusion by multiple rather than single occluder(s)
- Extension of image-space
  point-based occlusion culling

Very related work - Fuzzy visibility

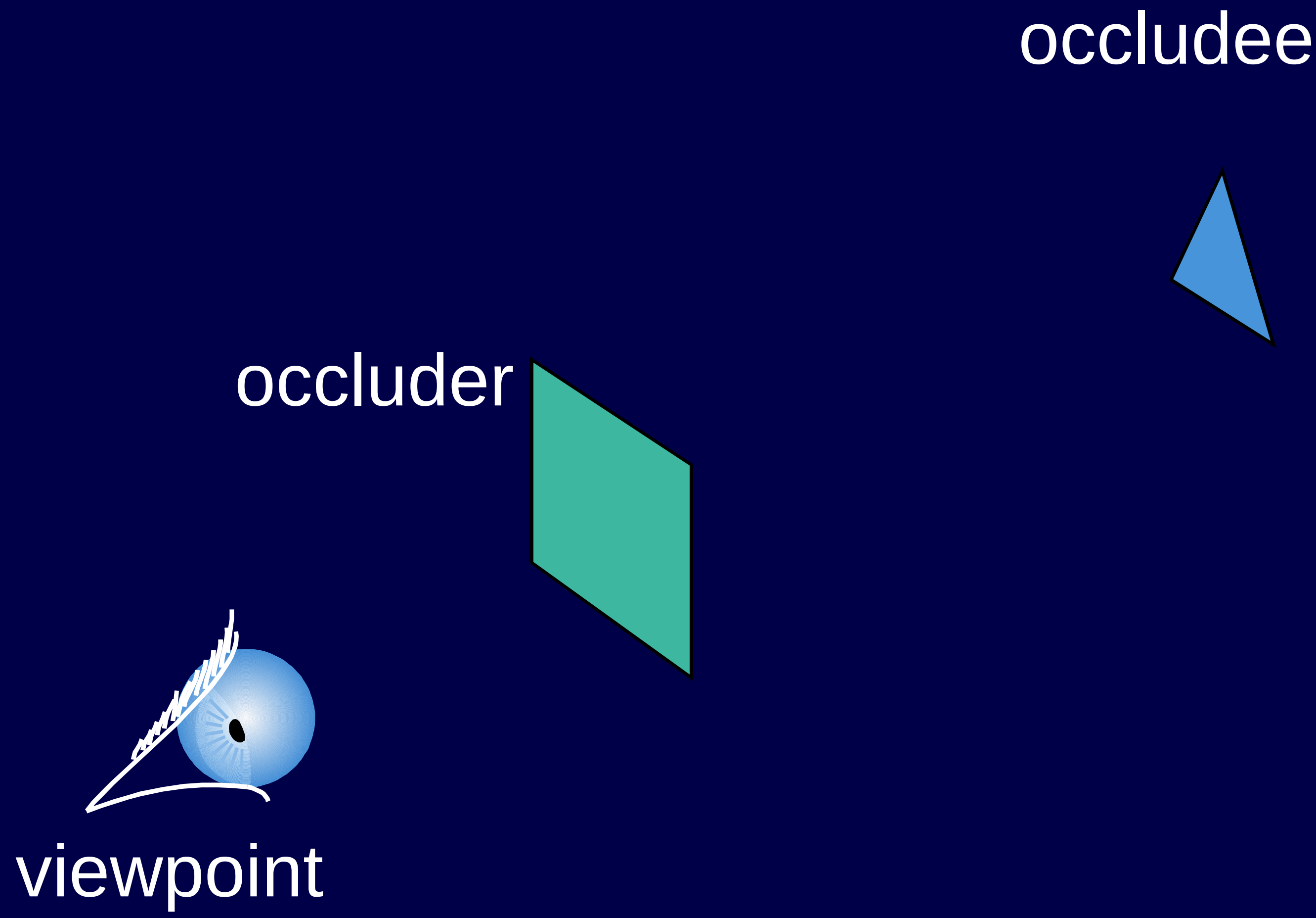
- Similar initial idea as ours
- Unfortunately unknown to us for final version

- [*Toward a Fuzzy Hidden Surface Algorithm*.
  Hong Lip Lim
  Computer Graphics International, Tokyo, 1992]

- Read the updated version of our paper
  http://graphics.lcs.mit.edu/~fredo

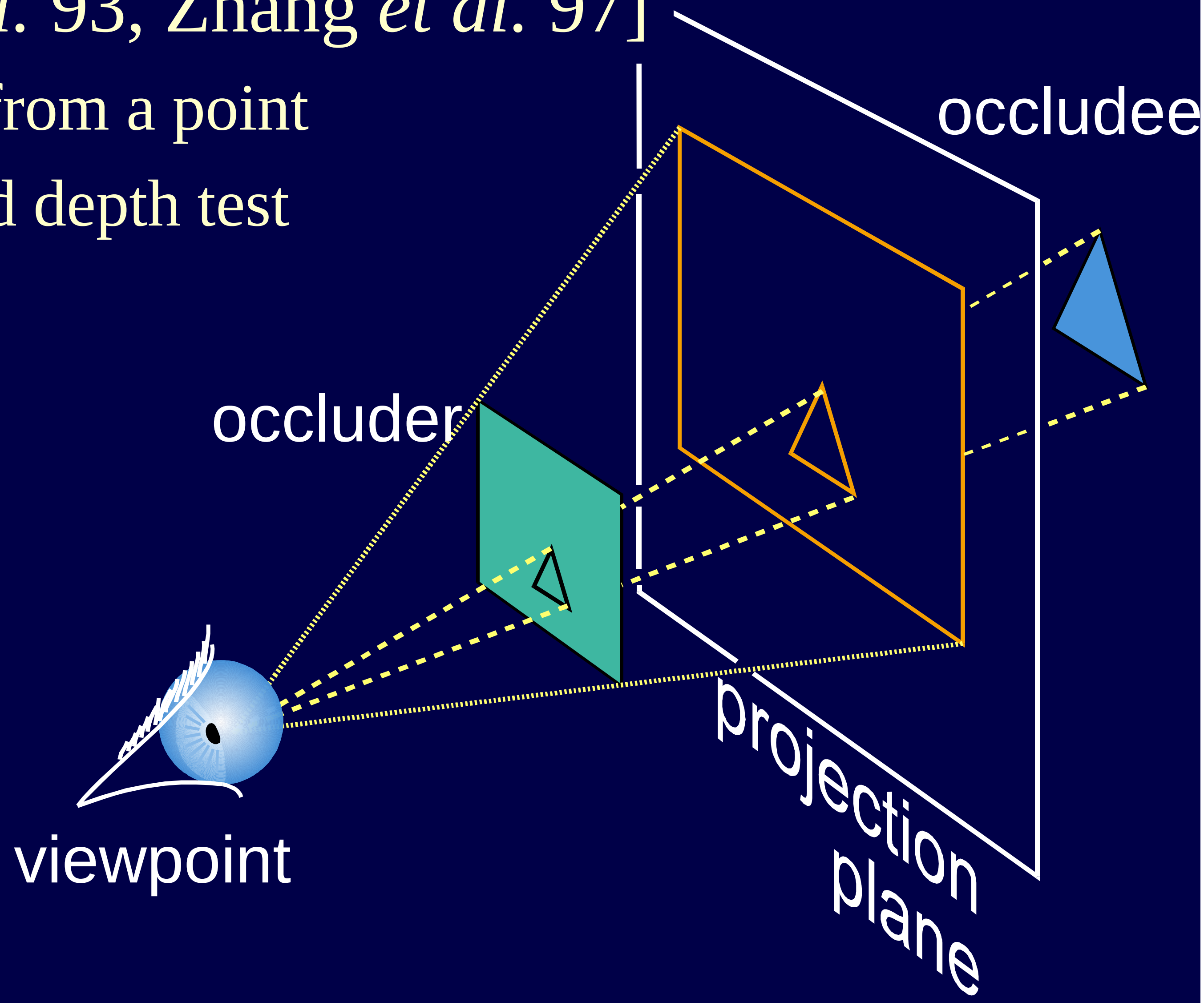
On-line point-based occlusion culling

- [Greene *et al*. 93, Zhang *et al*. 97]



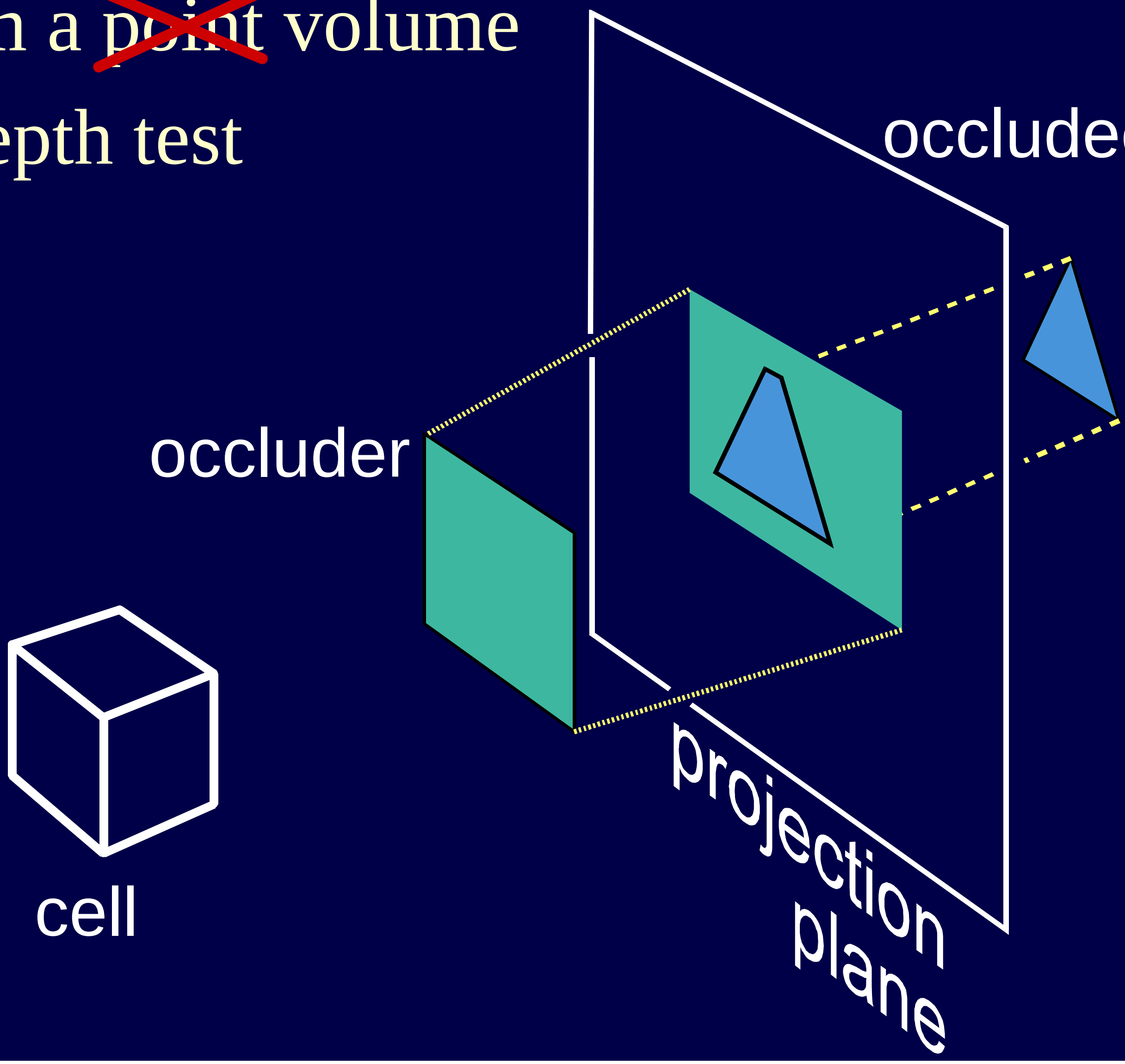
On-line point-based occlusion culling

- [Greene *et al.* 93, Zhang *et al.* 97]
  - Projection from a point
  - Overlap and depth test



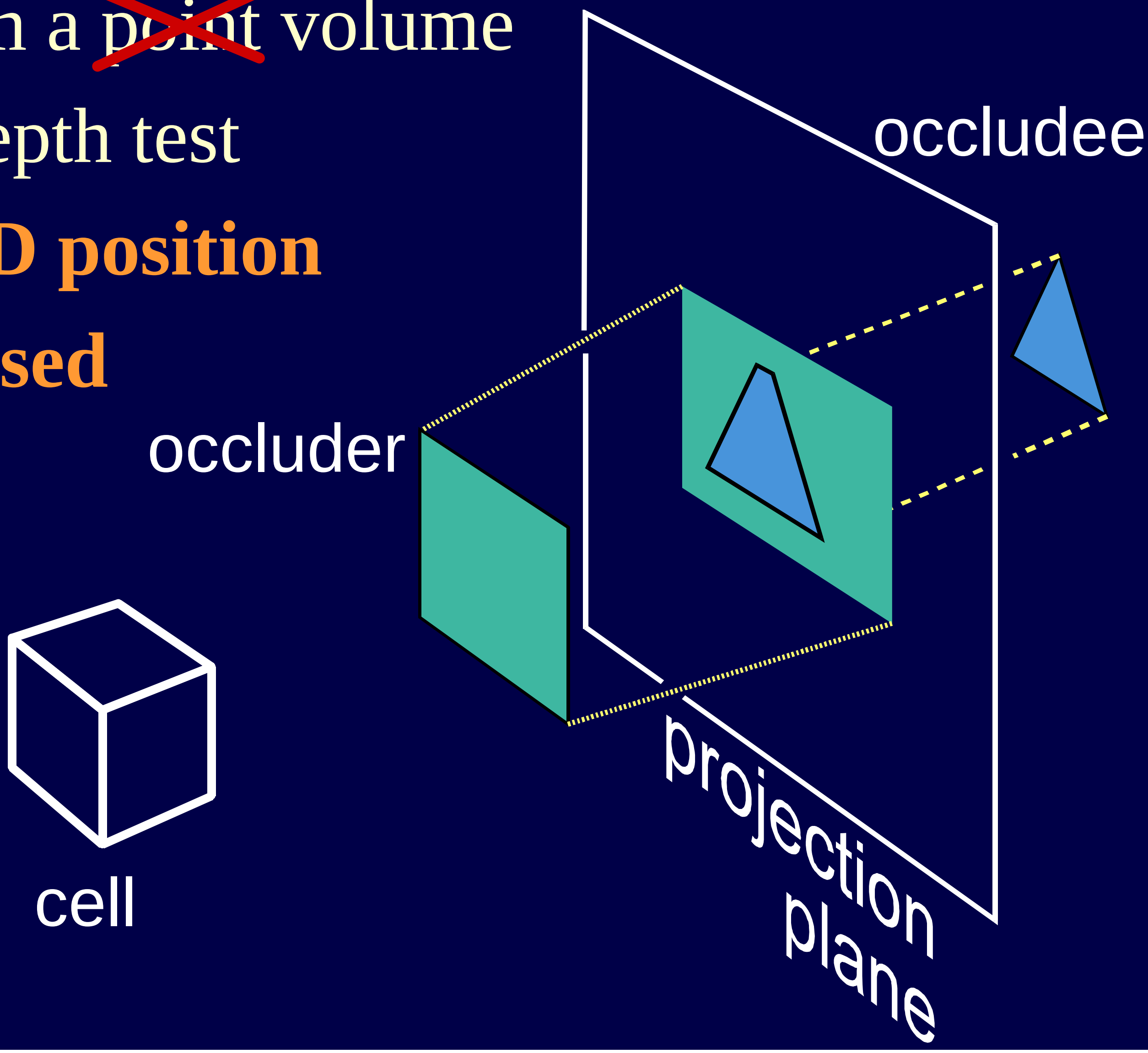
Extended projections

- Projection from a ~~point~~ volume
- Overlap and depth test



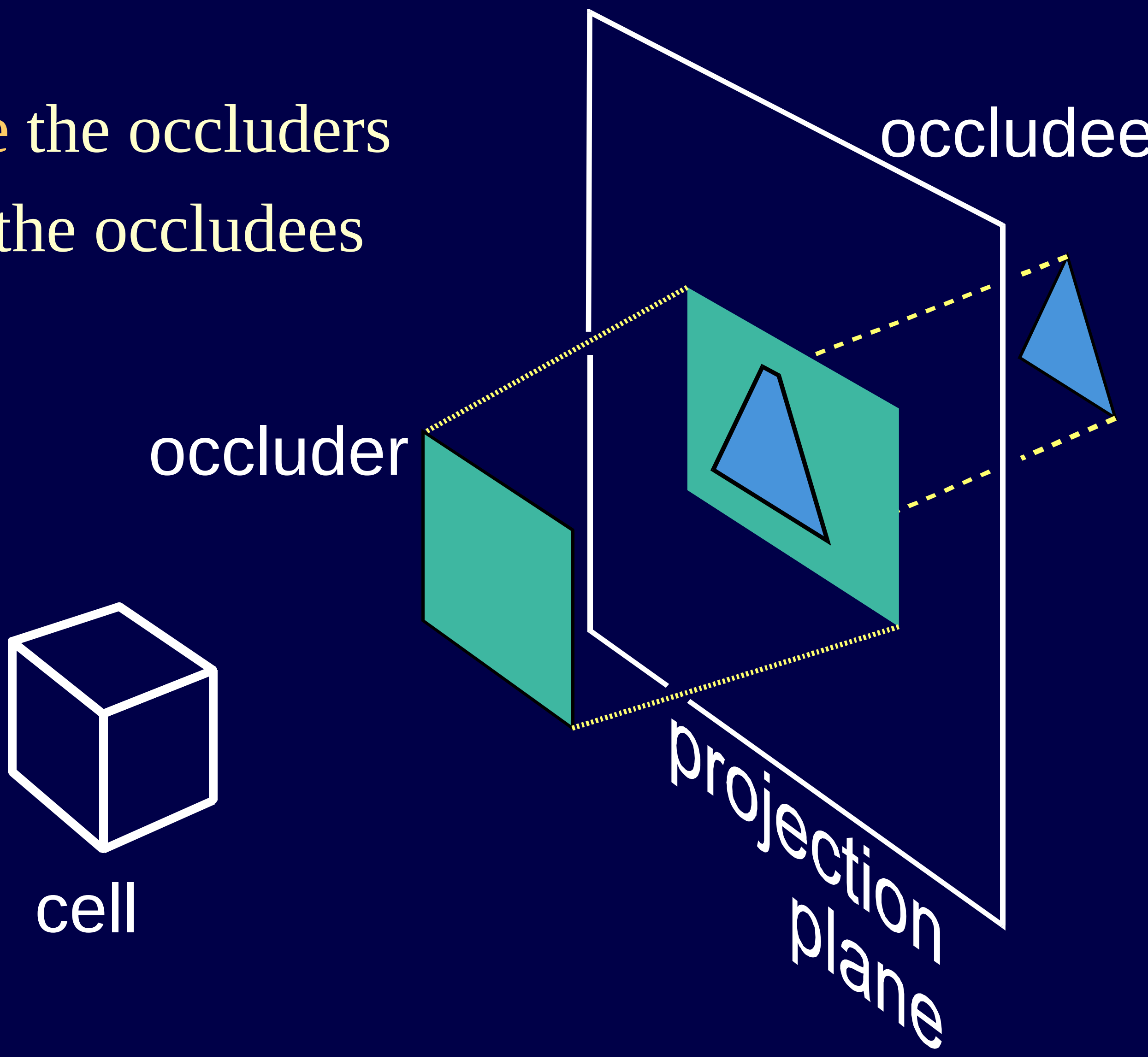
Extended projections

- Projection from a ~~point~~ volume
- Overlap and depth test
- **Fixed plane 3D position**
- **Will be discussed**



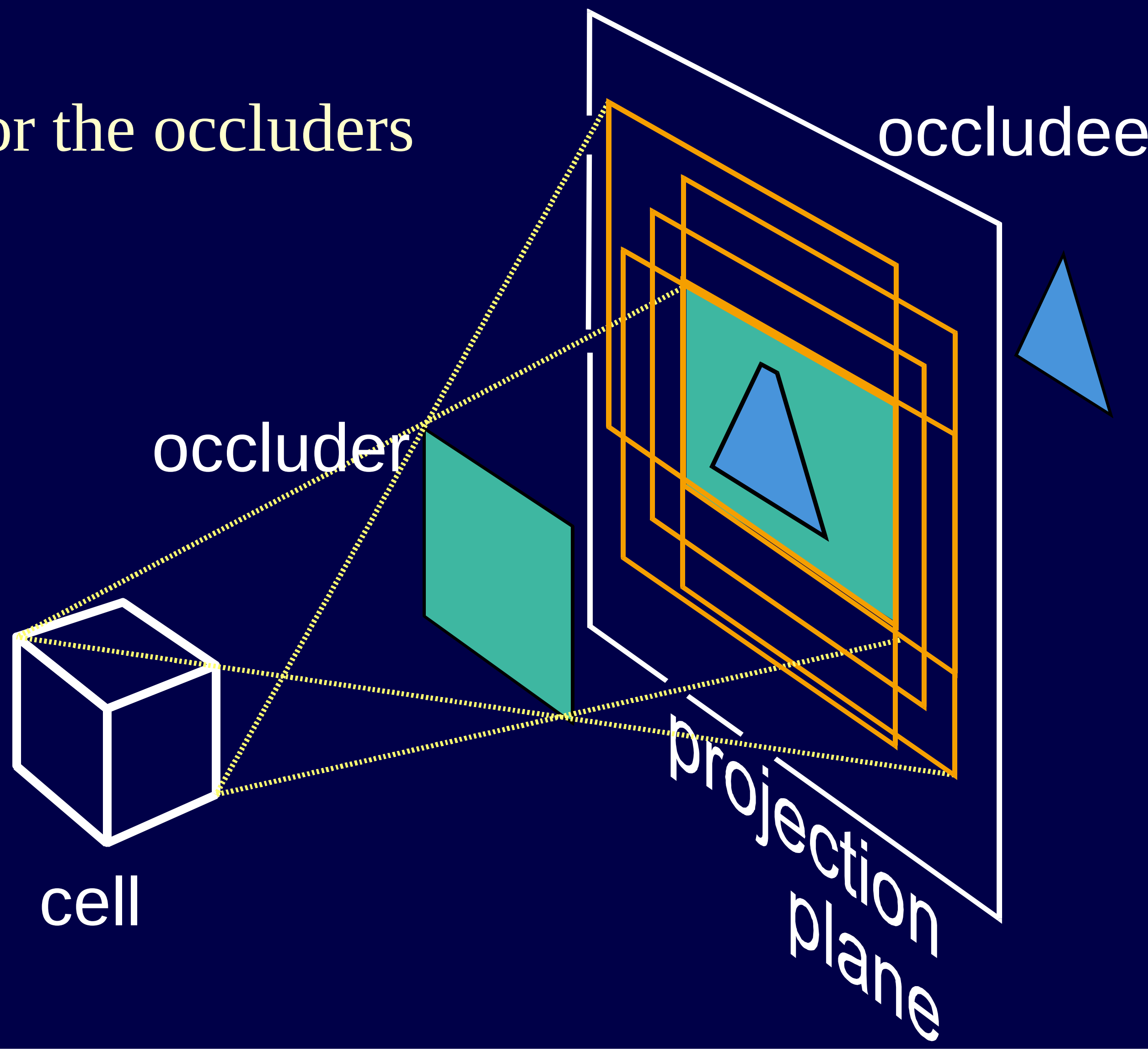
Extended projections

- Conservative
  - Underestimate the occluders
  - Overestimate the occludees



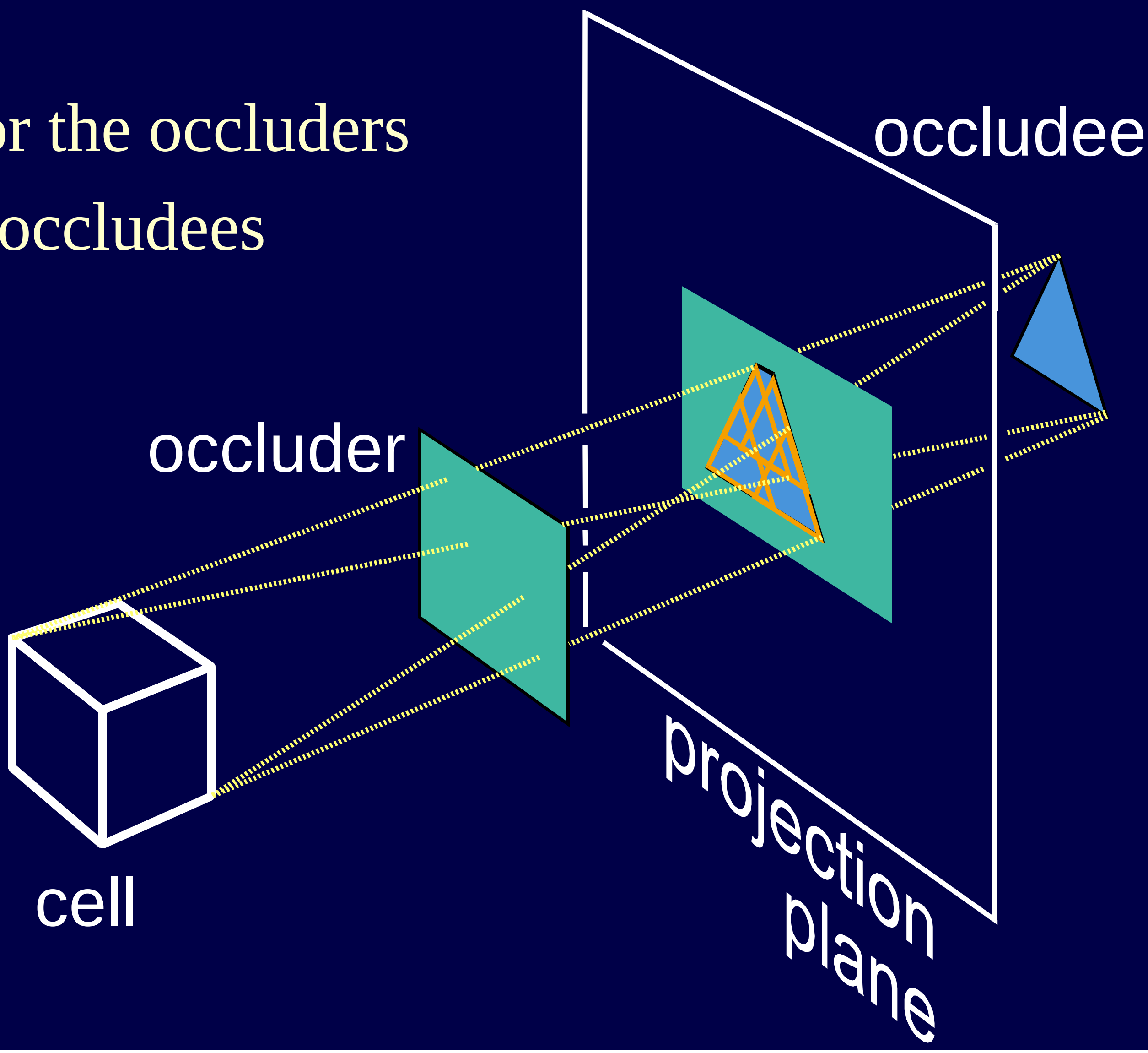
Extended projections

- Conservative
  - Intersection for the occluders



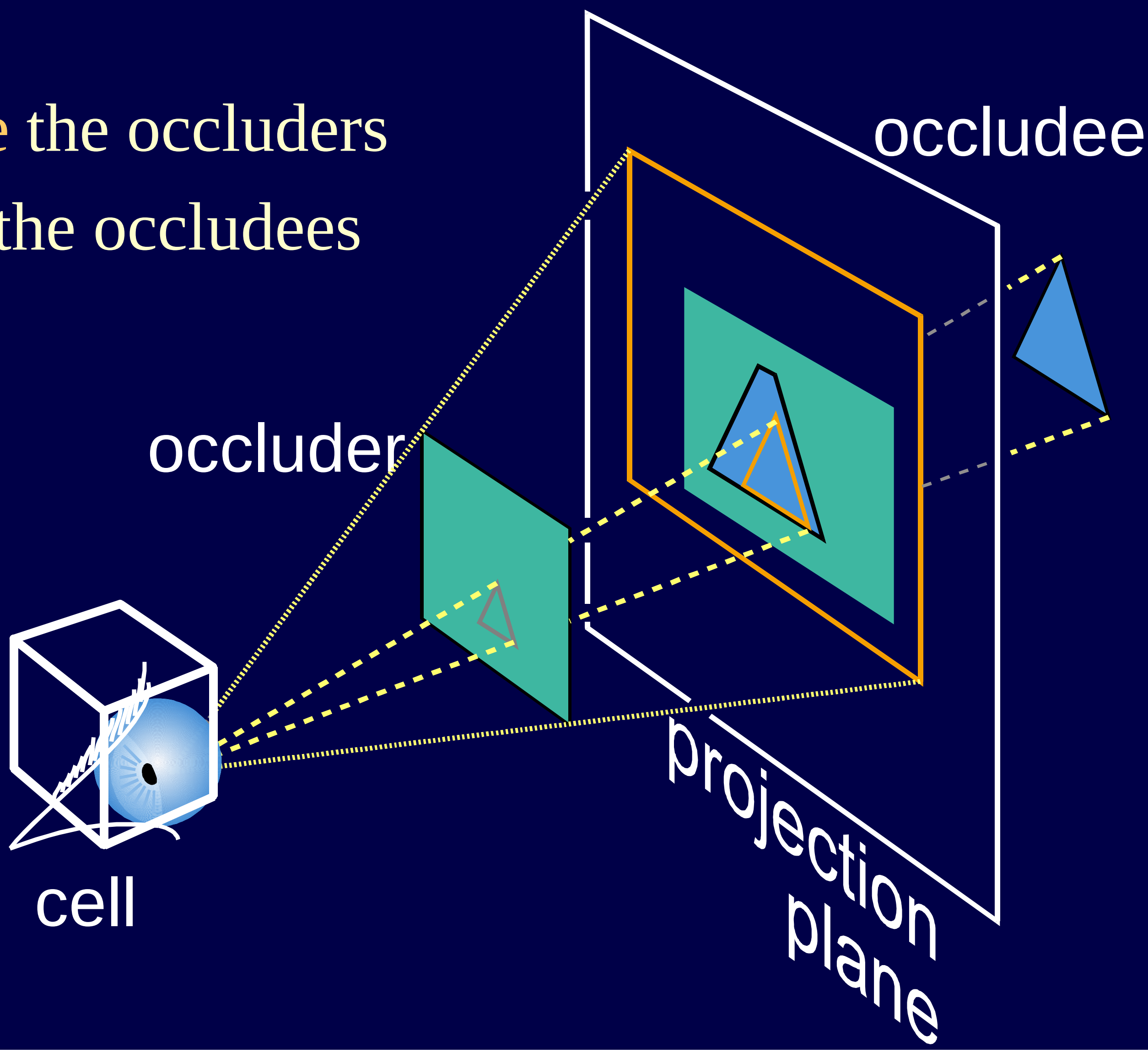
Extended projections

- Conservative
  - Intersection for the occluders
  - Union for the occludees



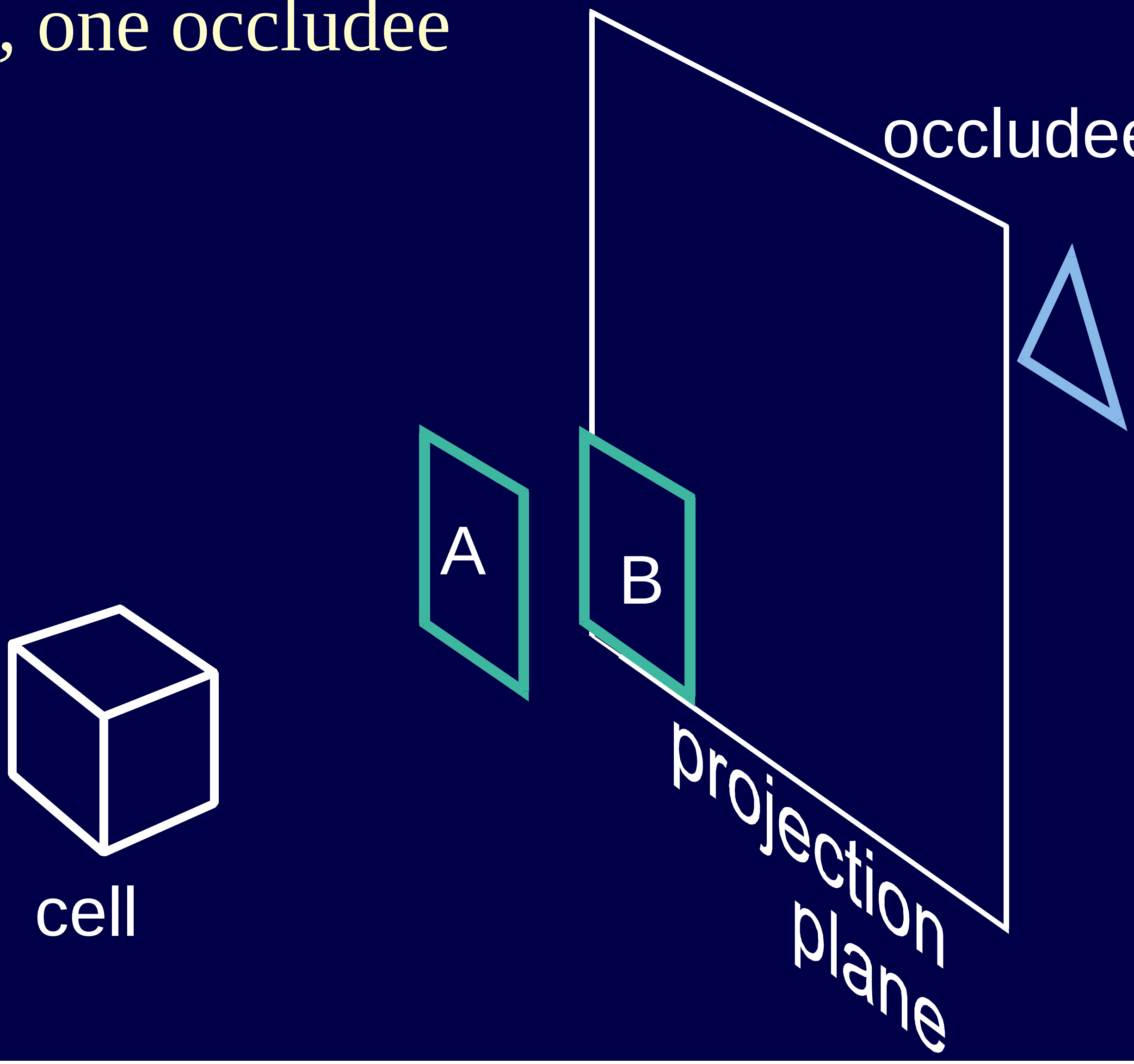
Extended projections

- Conservative
  - Underestimate the occluders
  - Overestimate the occludees



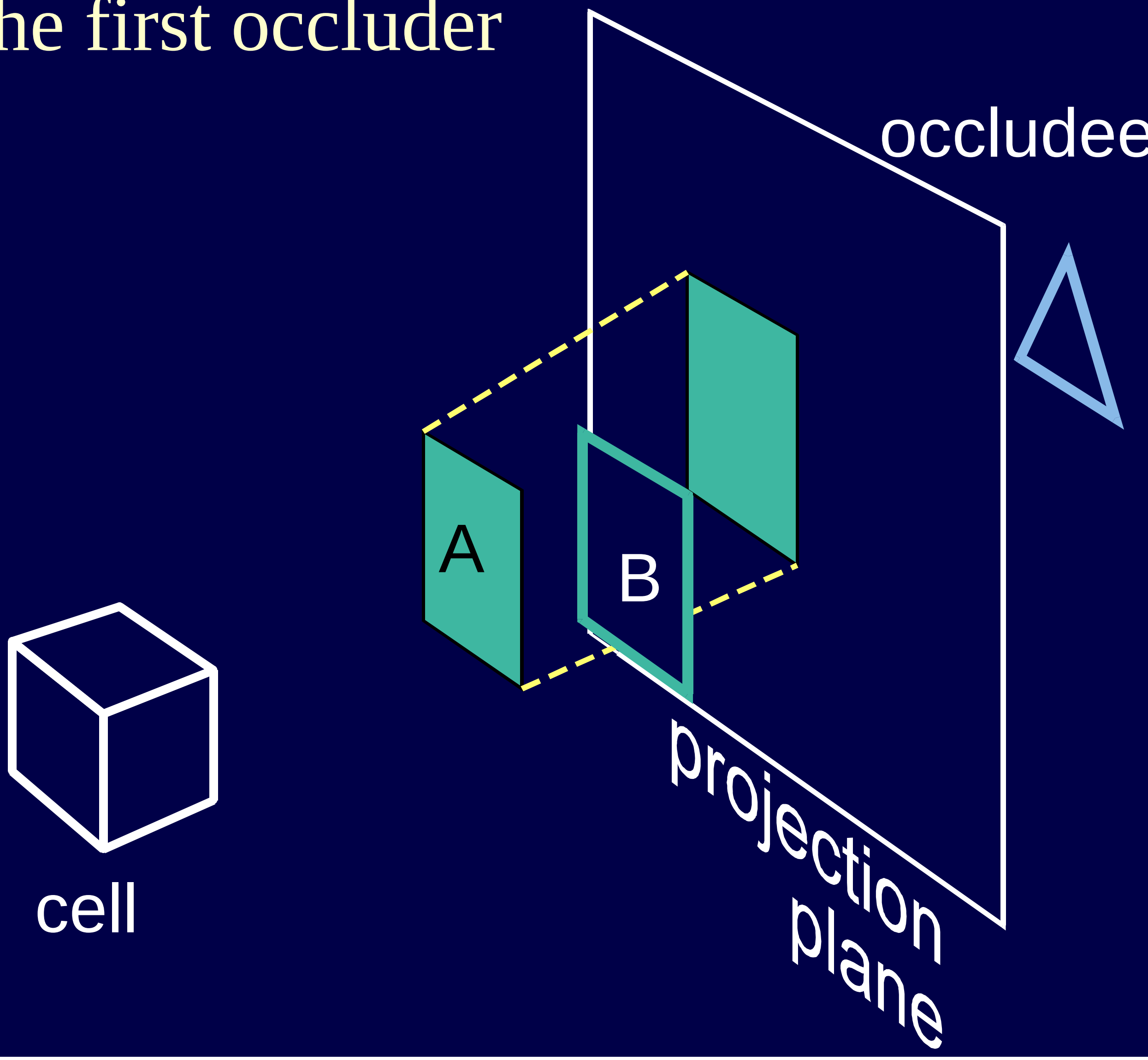
Occluder fusion

- Two occluders, one occludee



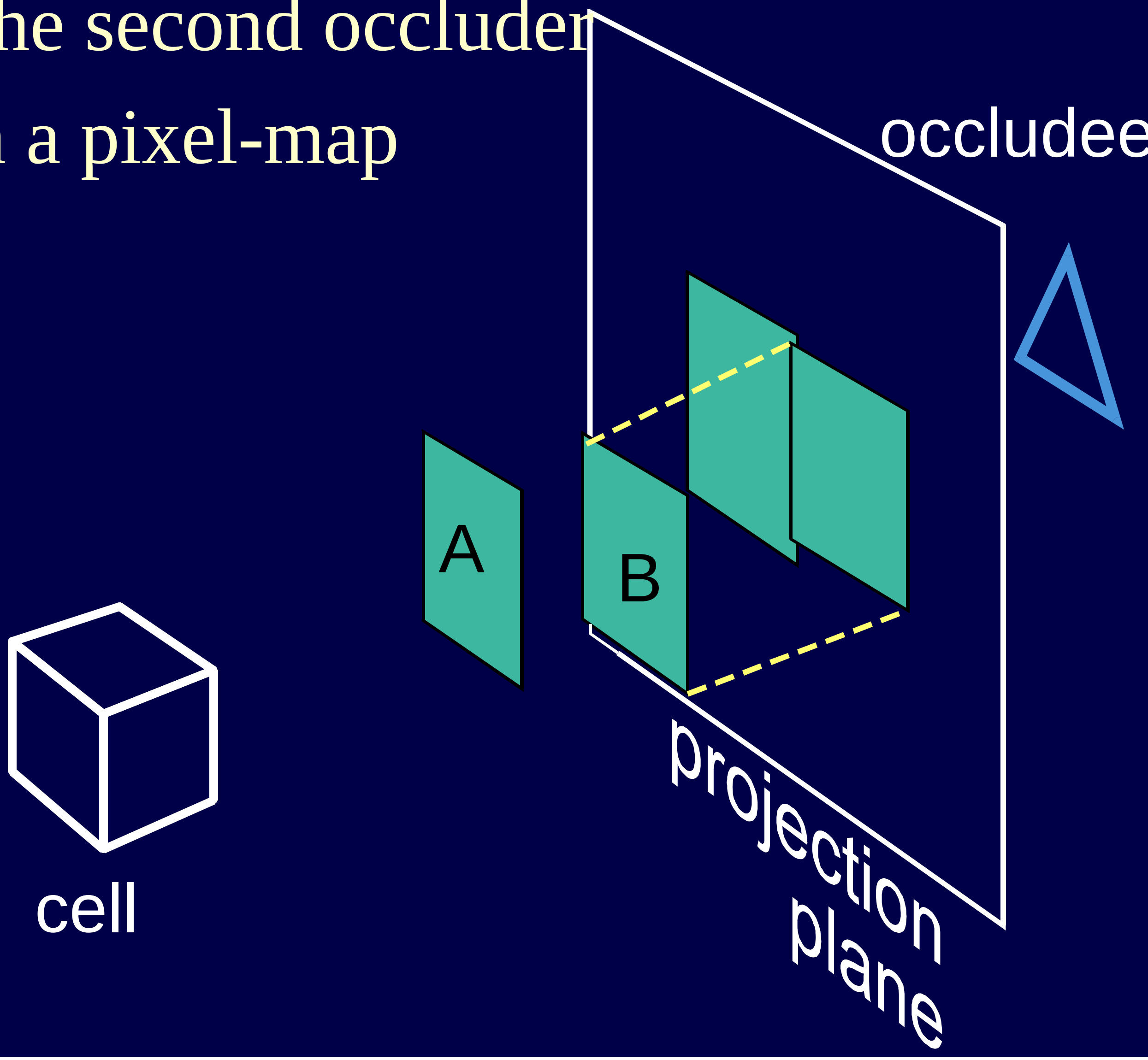
Occluder fusion

- *Projection* of the first occluder



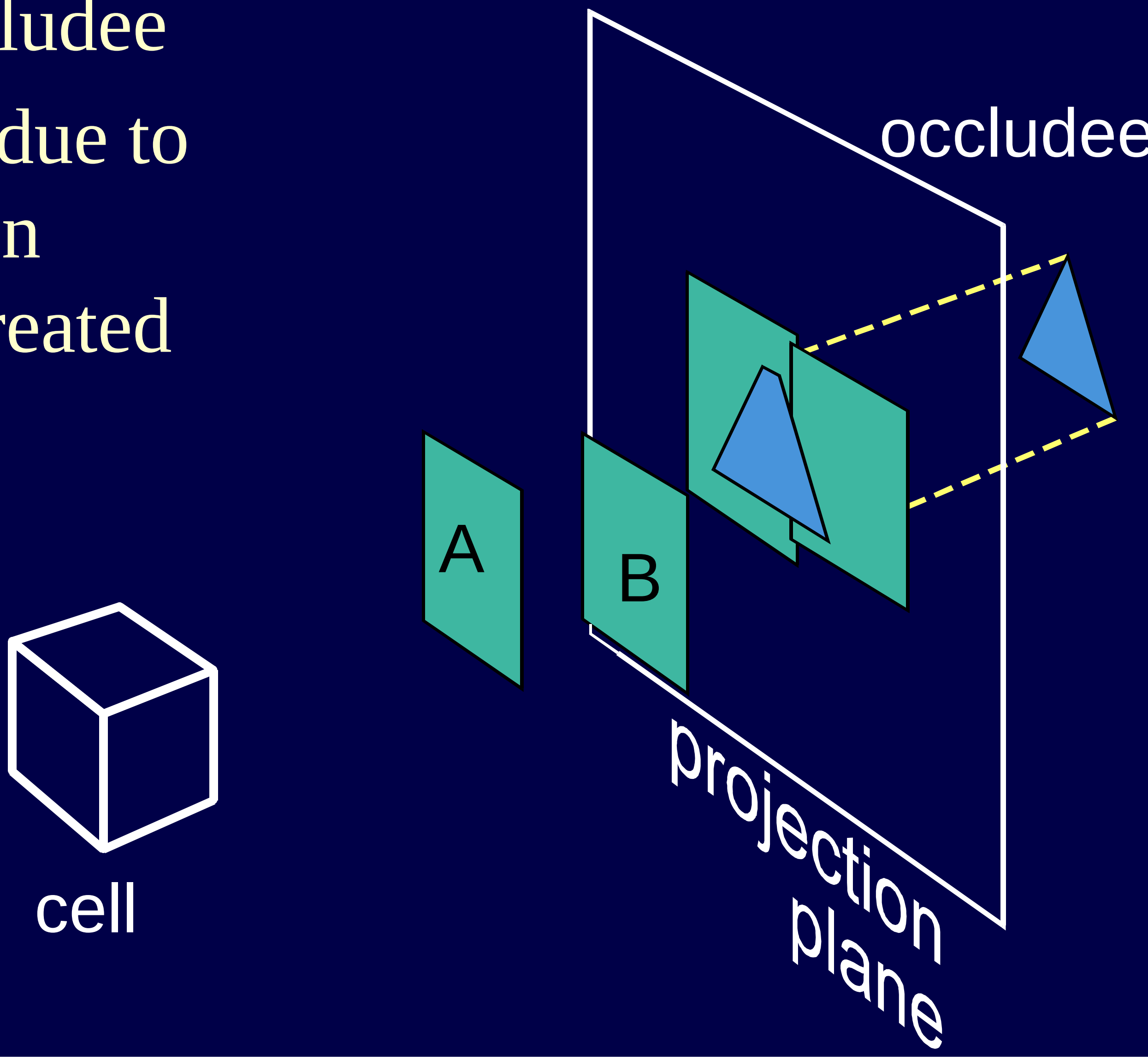
Occluder fusion

- *Projection* of the second occluder
- Aggregation in a pixel-map



Occluder fusion

- Test of the occludee
- The occlusion due to the combination of *A* and *B* is treated



Fuzzy visibility

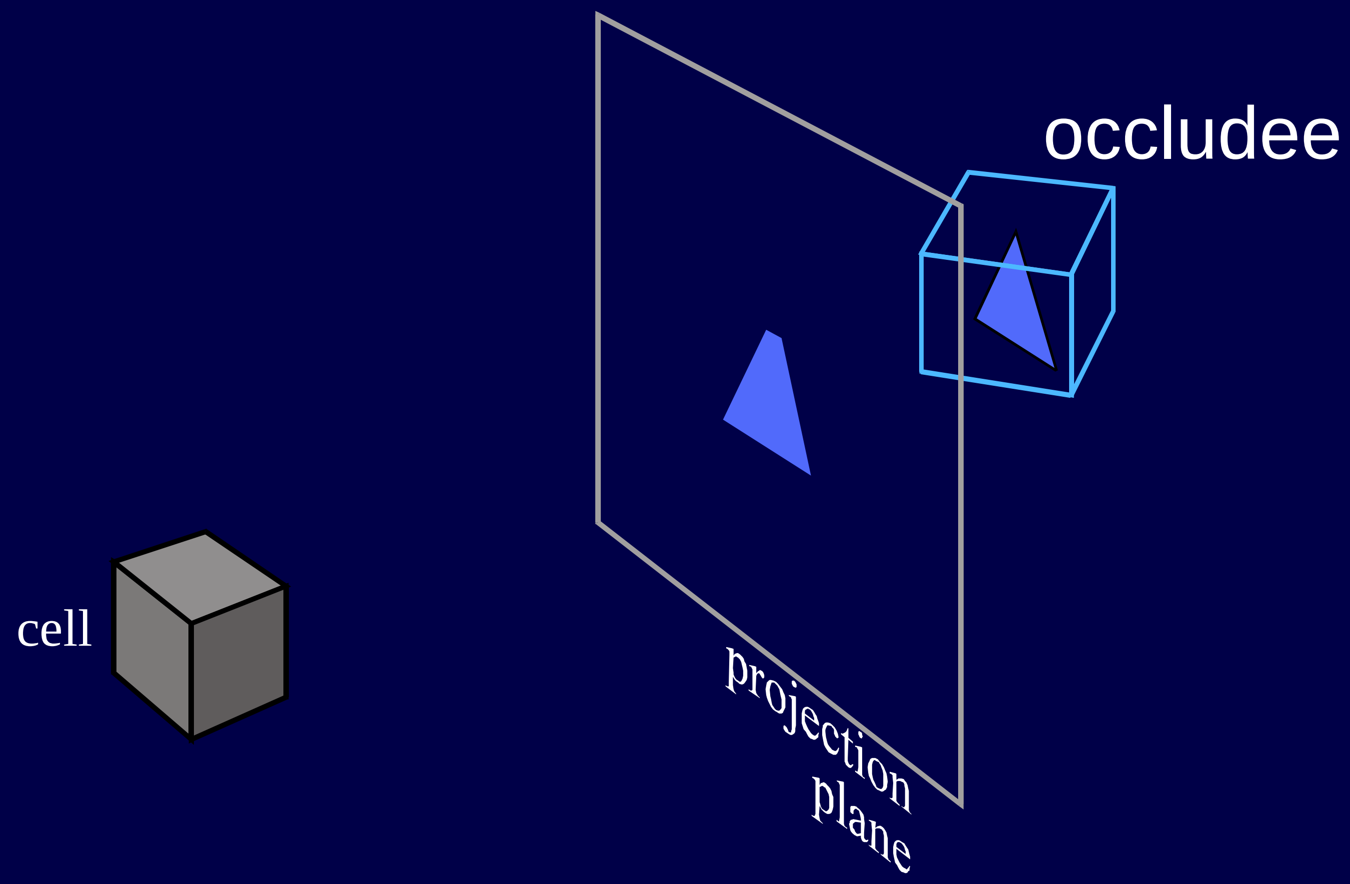
[Lim 1992]

- Extended projection as a fuzzy analysis
- Same definition with unions/intersections
- However, plane at infinity (direction space)
  - Thus works only for infinite umbra
- Concave mesh projection

Our new method

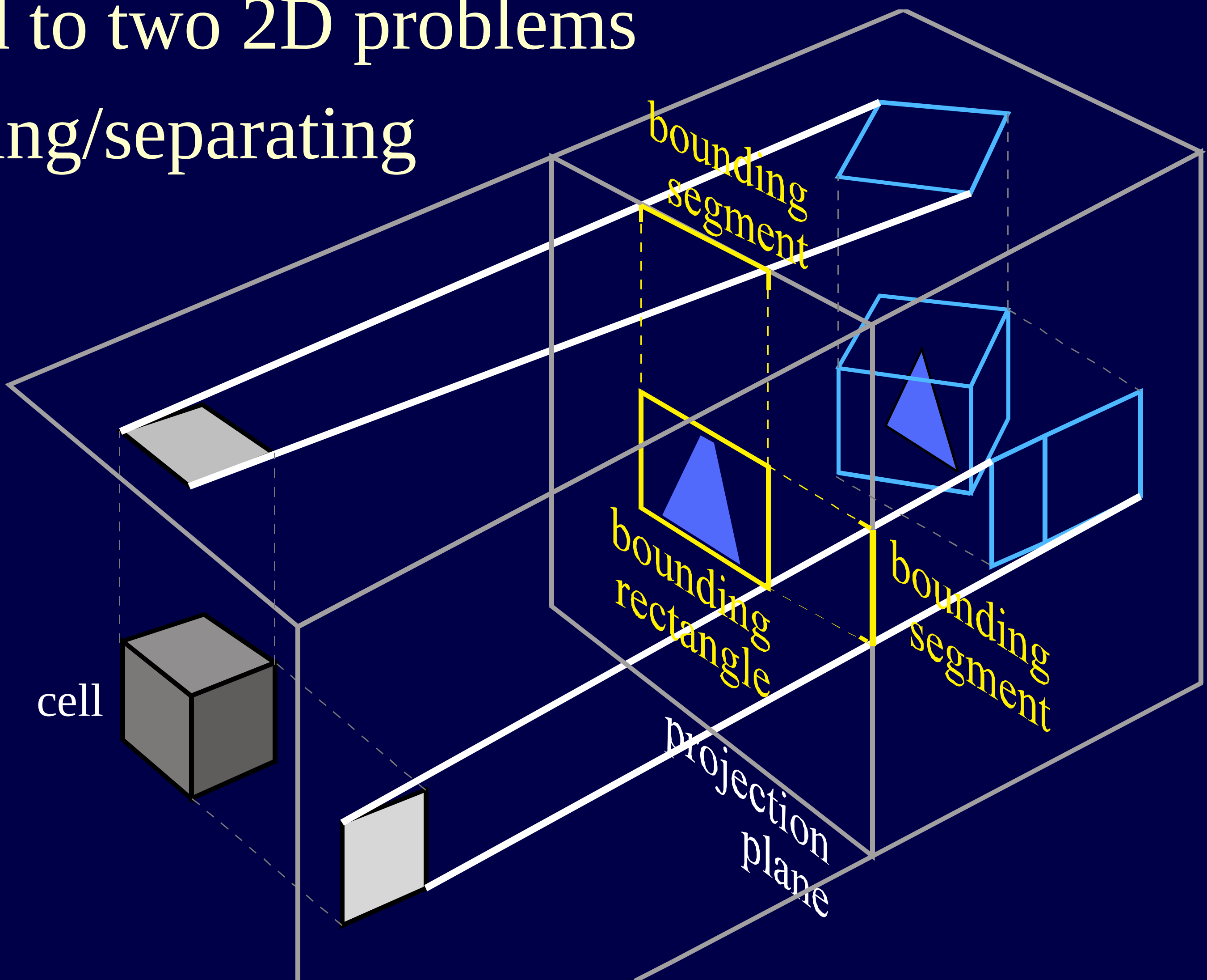
- New Projection algorithms
- Heuristic for choice of projection plane
- Reprojection
- Occlusion sweep
- Improved projection
- Occlusion culling system

Occludee Projection



Occludee Projection

- Reduced to two 2D problems
- Supporting/separating lines

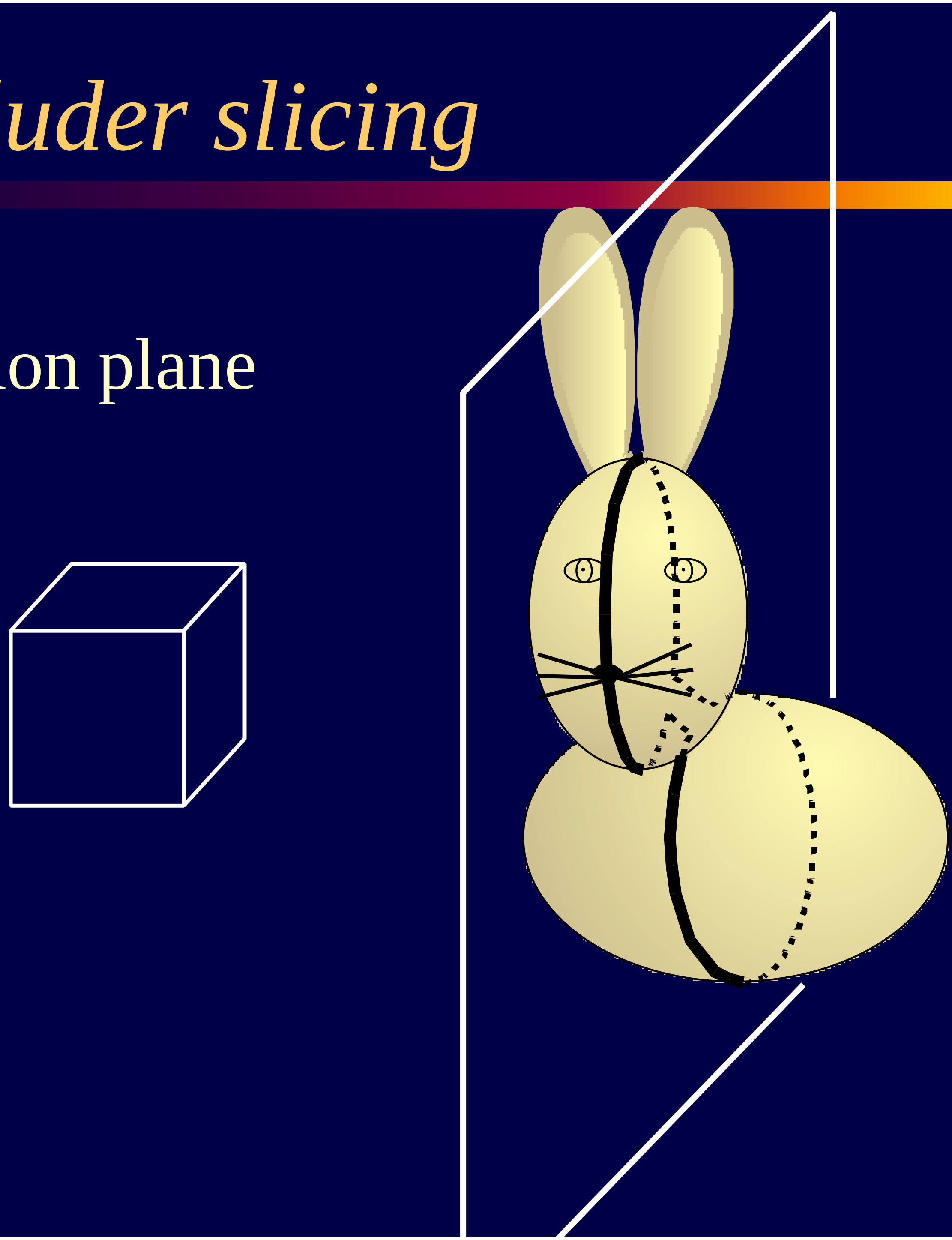


Convex occluder Projection

- Convex cell =>
  intersection of views from vertices of the cell
- Hardware computation using the stencil buffer
- Conservative rasterization

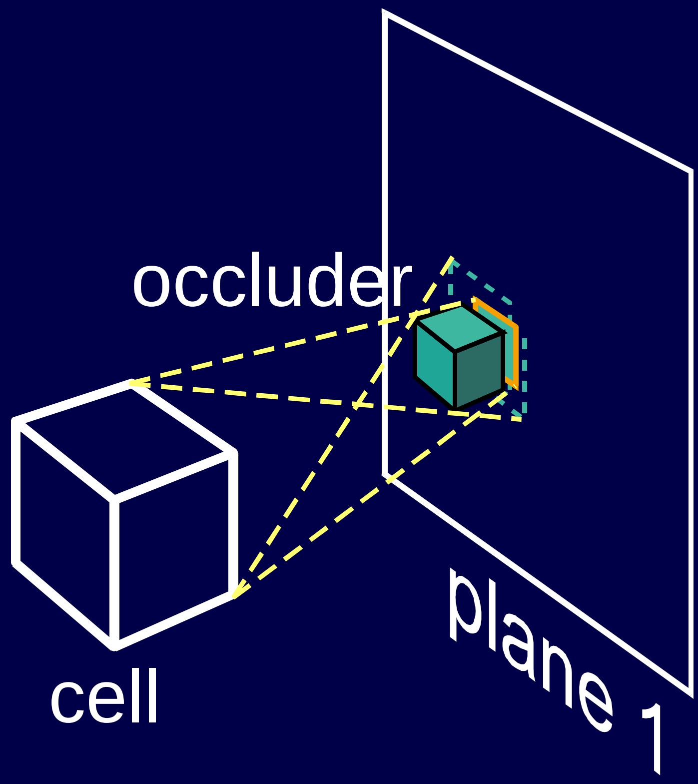
Concave occluder slicing

- Intersection
  occluder-projection plane



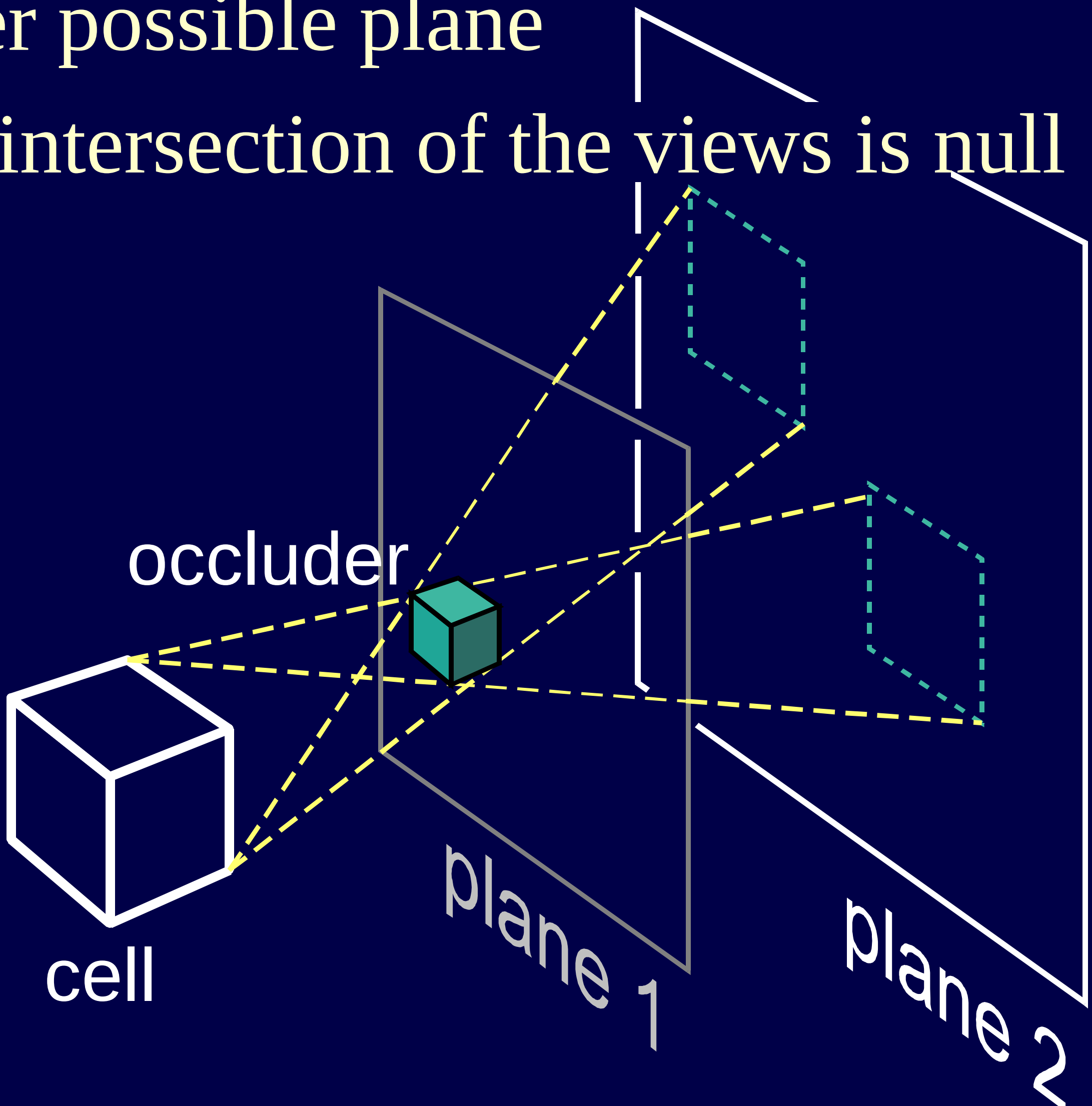
Difficulty of choosing the plane

- First possible plane
- Fine



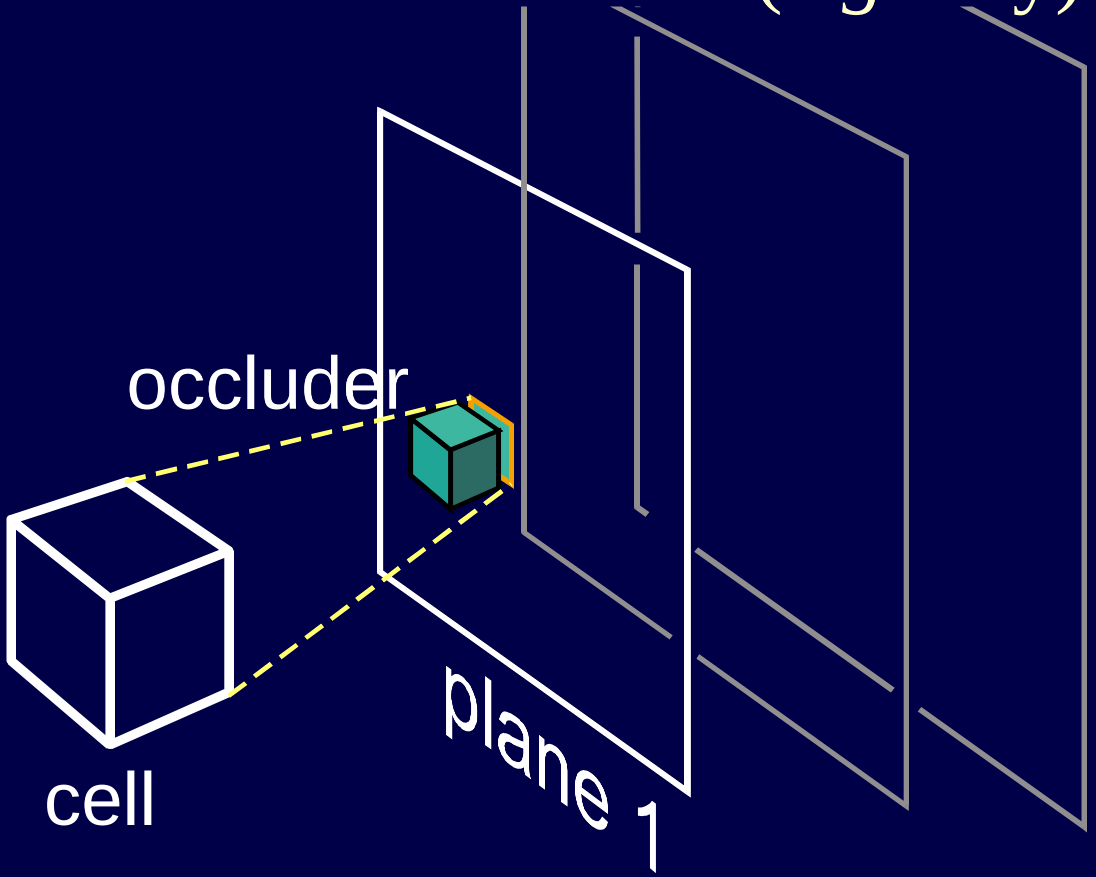
Difficulty of choosing the plane

- Other possible plane
- The intersection of the views is null



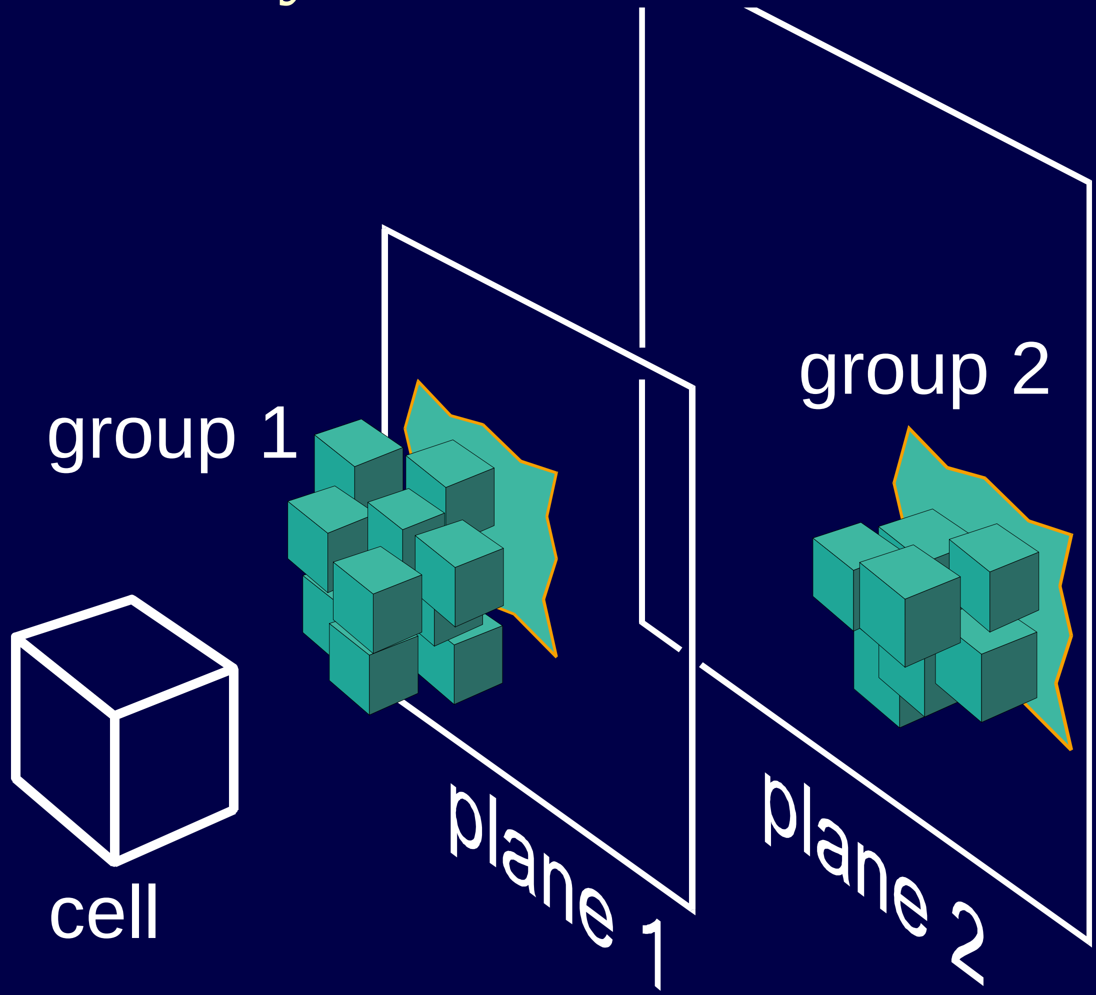
Choosing the plane

- Heuristic (maximize projected surface)
- Works fine for most cases (*e.g.* city)



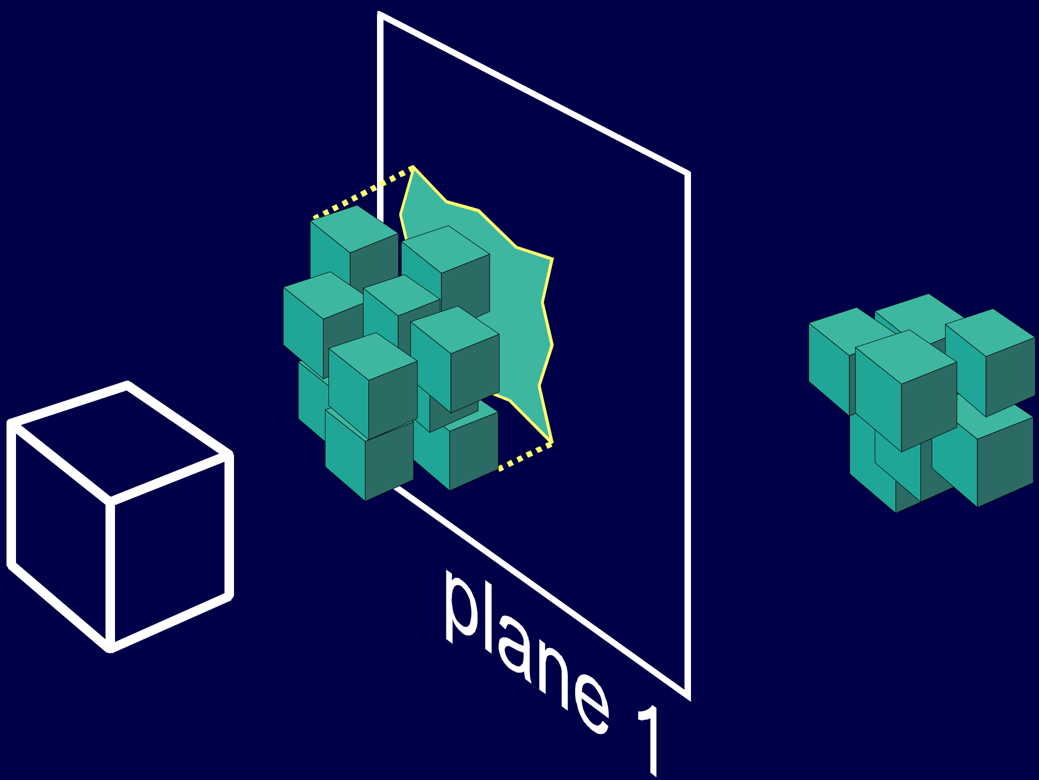
Problem of the choice of the plane

- Contradictory constraints



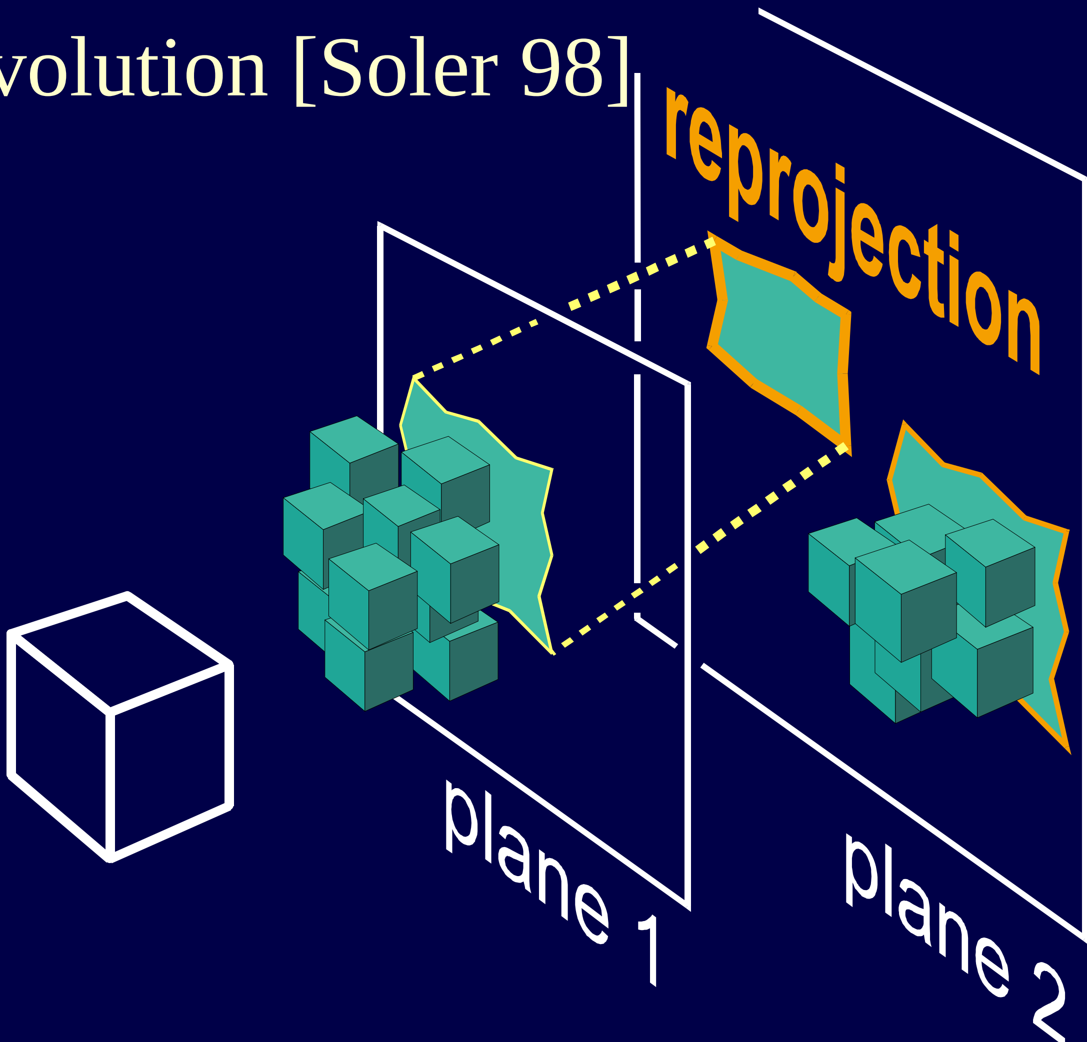
Solution

- Project on plane 1
  - Aggregate extended projections



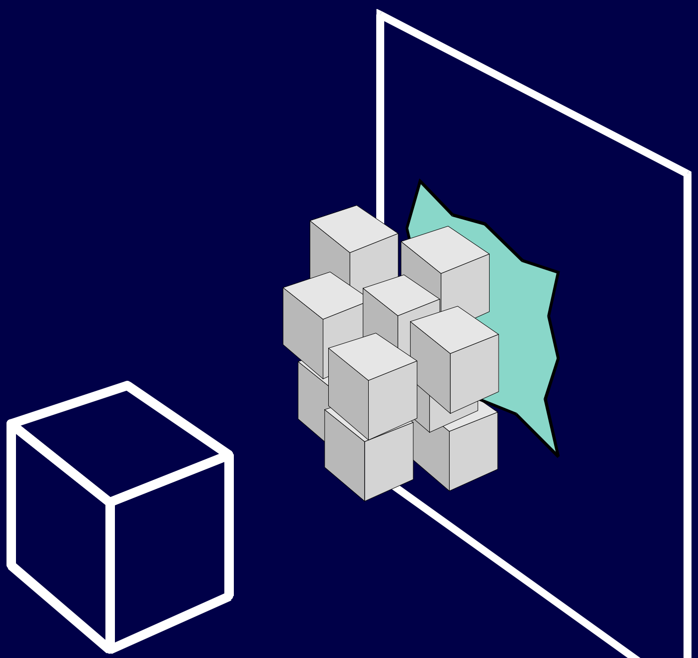
Re-projection

- Re-project aggregated occlusion map onto plane 2
- Convolution [Soler 98]



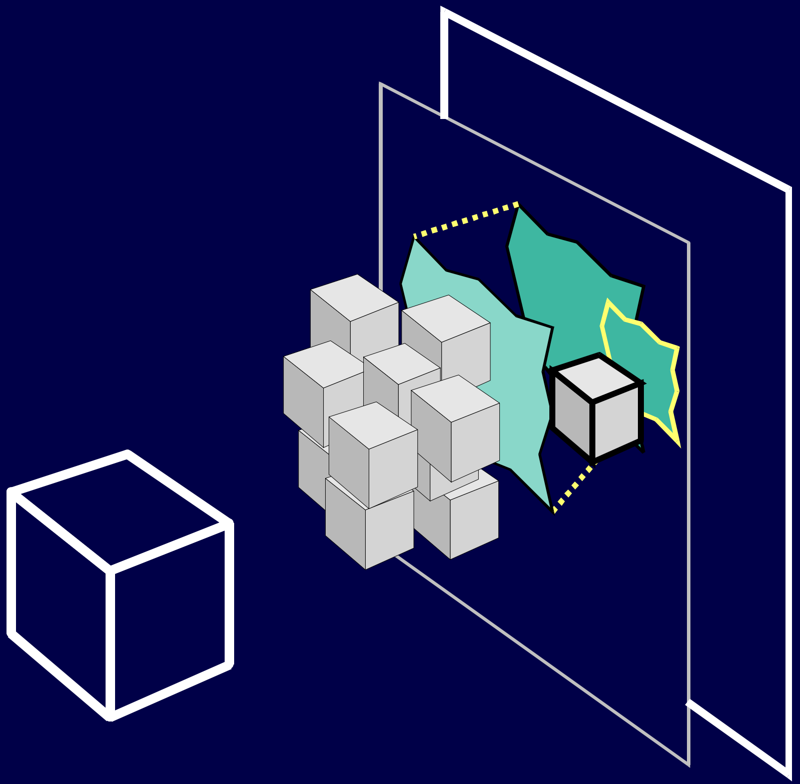
Occlusion sweep

- Initial projection plane



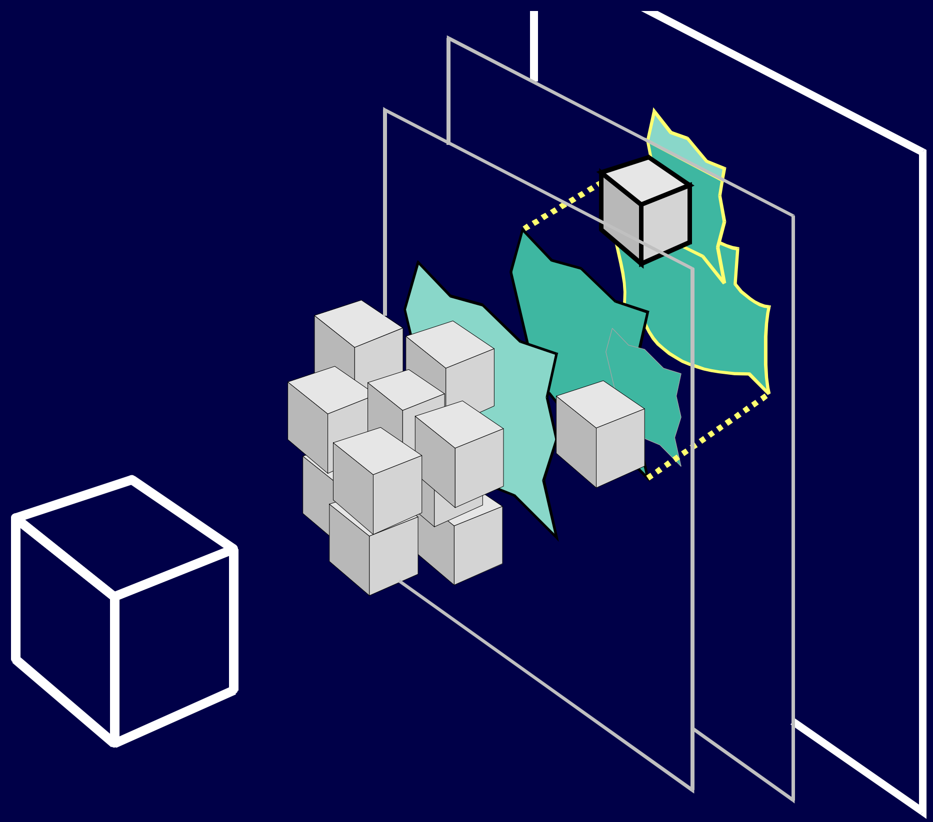
Occlusion sweep

- Re-projection
- *Projection* of new occluders



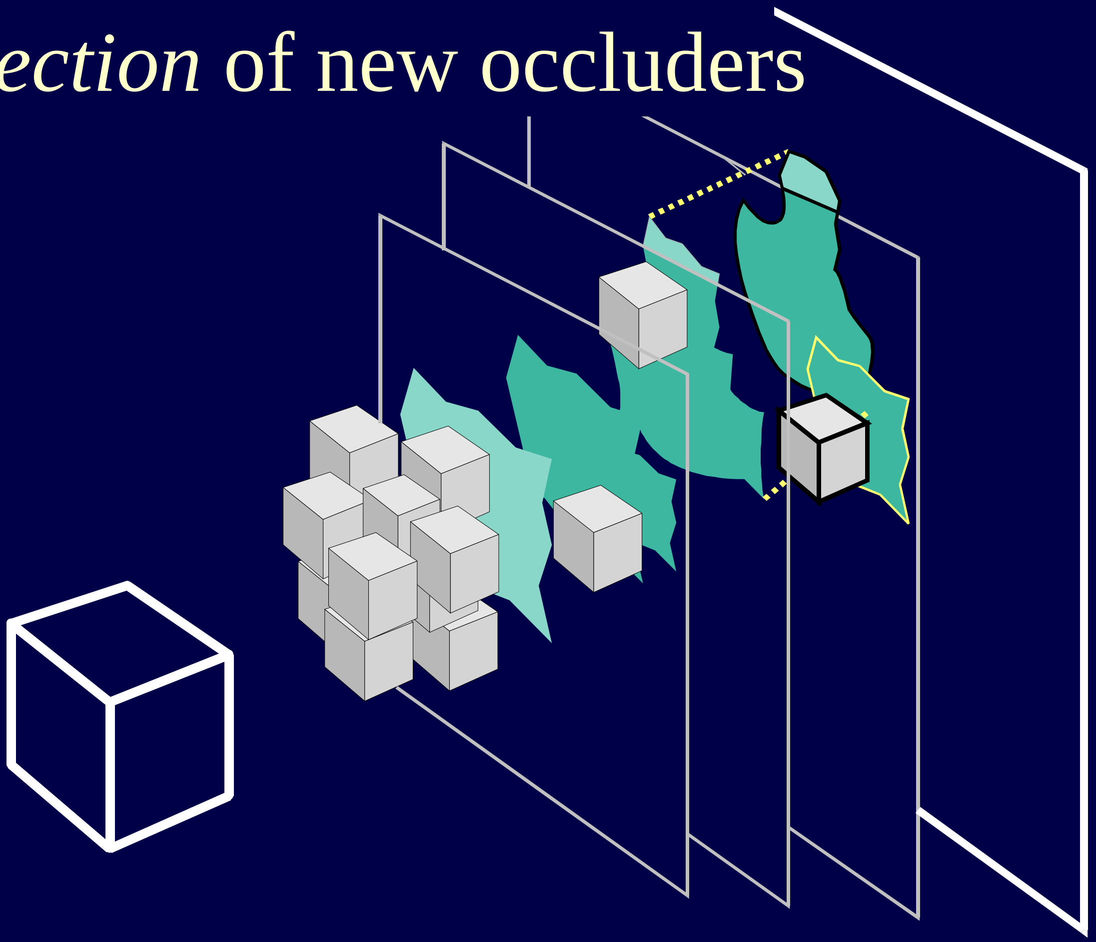
Occlusion sweep

- Re-projection
- *Projection* of new occluders



Occlusion sweep

- Re-projection
- *Projection* of new occluders

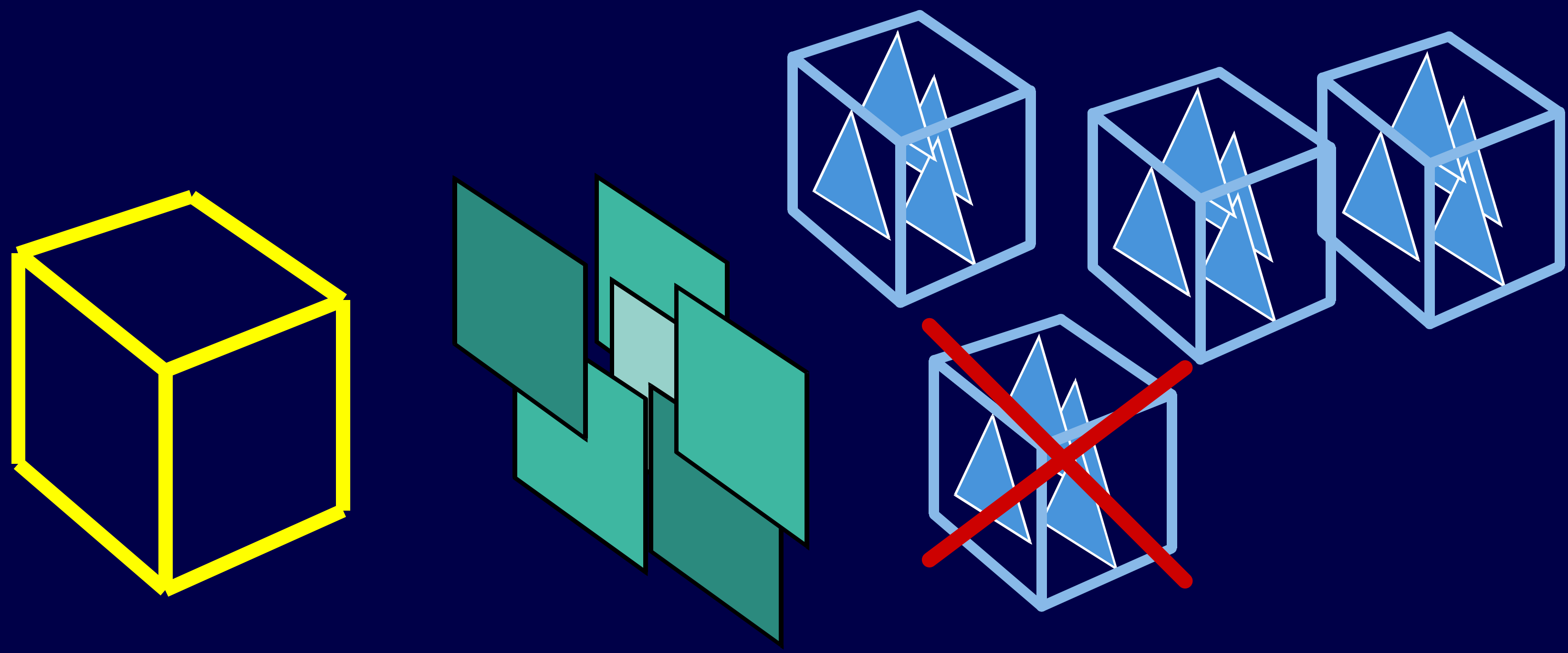


Improved Extended Projection

- Detect more occlusion for some configurations
- For convex and planar occluders
- Do not use unions for occludees
  (supporting lines only)

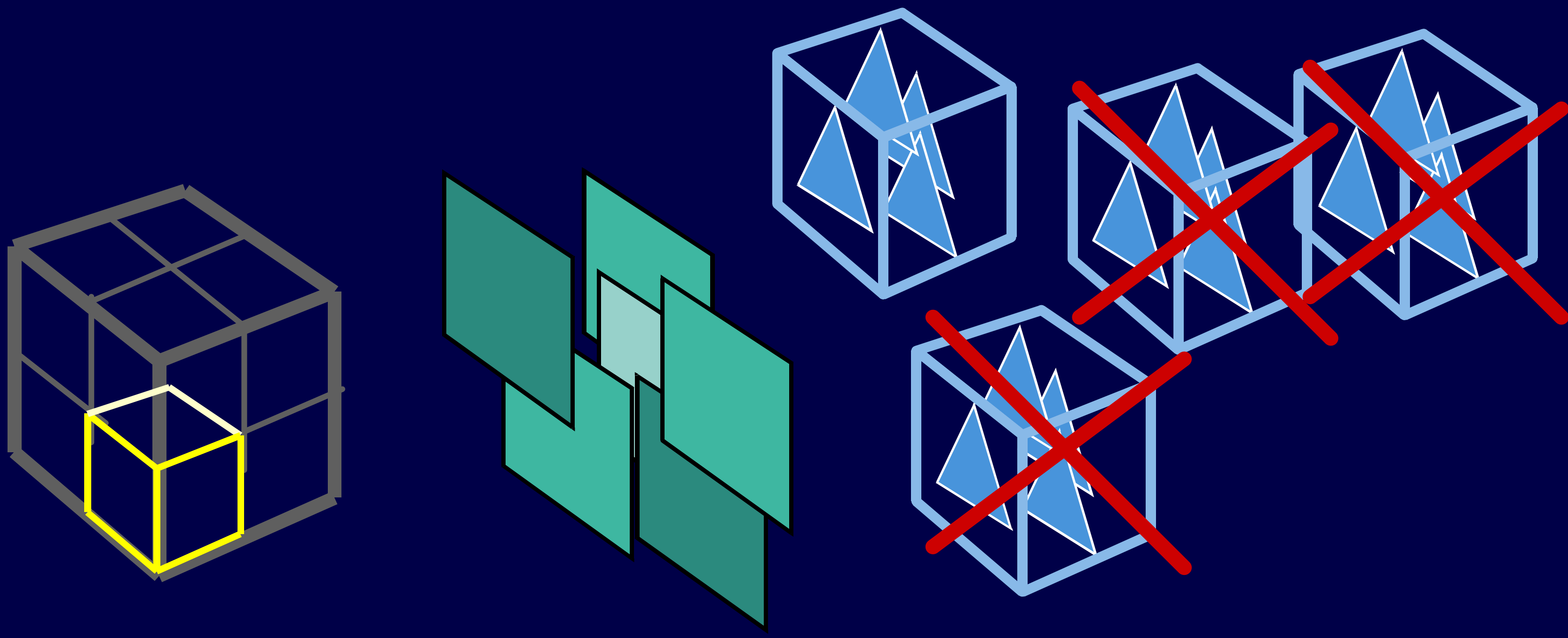
Adaptive preprocessing

- If cell has too many visible objects



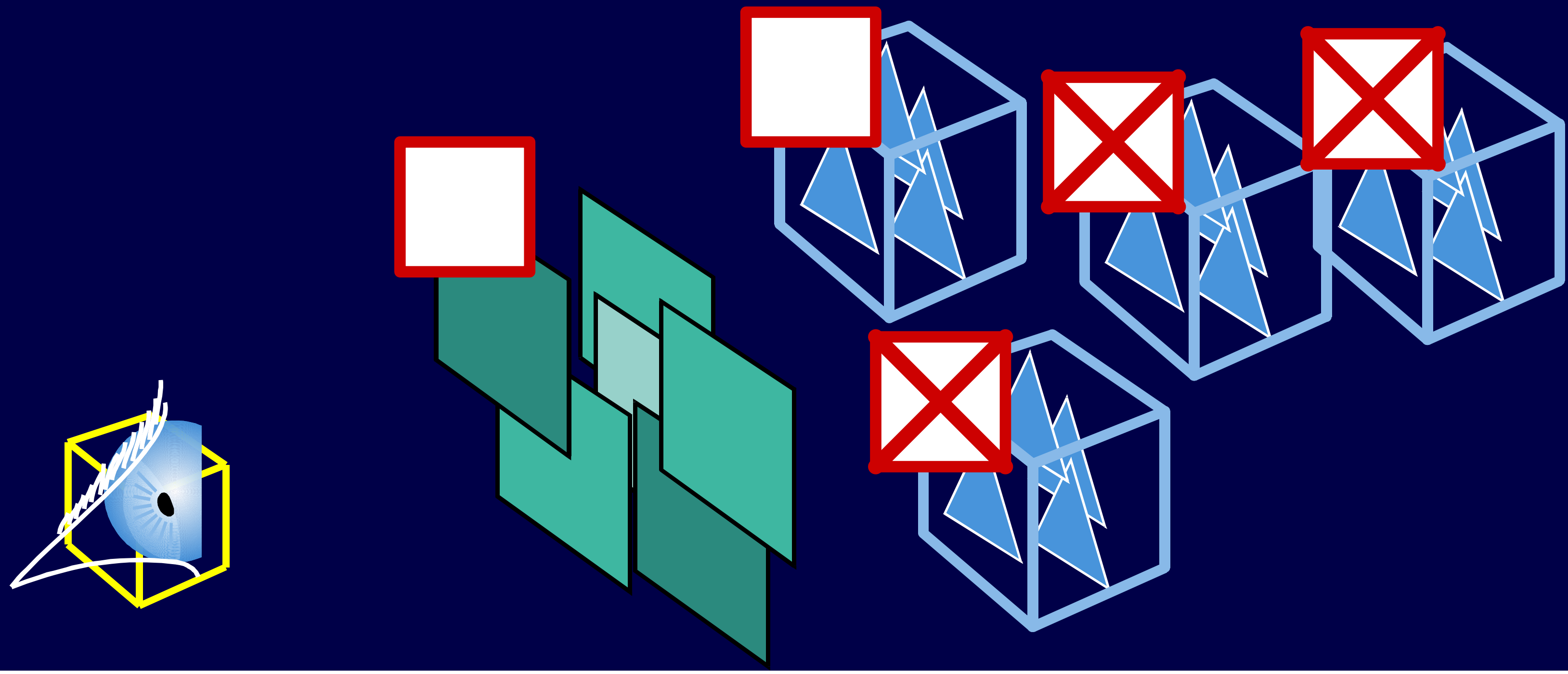
Adaptive preprocessing

- If cell has too many visible objects then subdivide



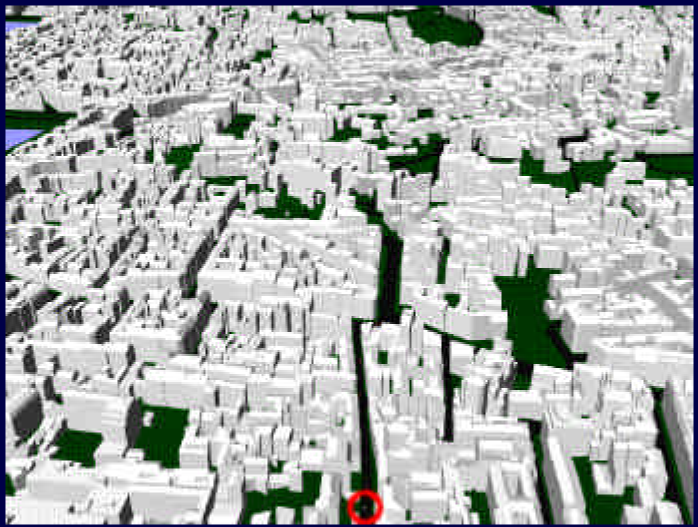
Interactive viewer

- Potentially Visible Set precomputation
- Visibility flag in the object hierarchy
- No cost at runtime
- Moving objects: motion volume

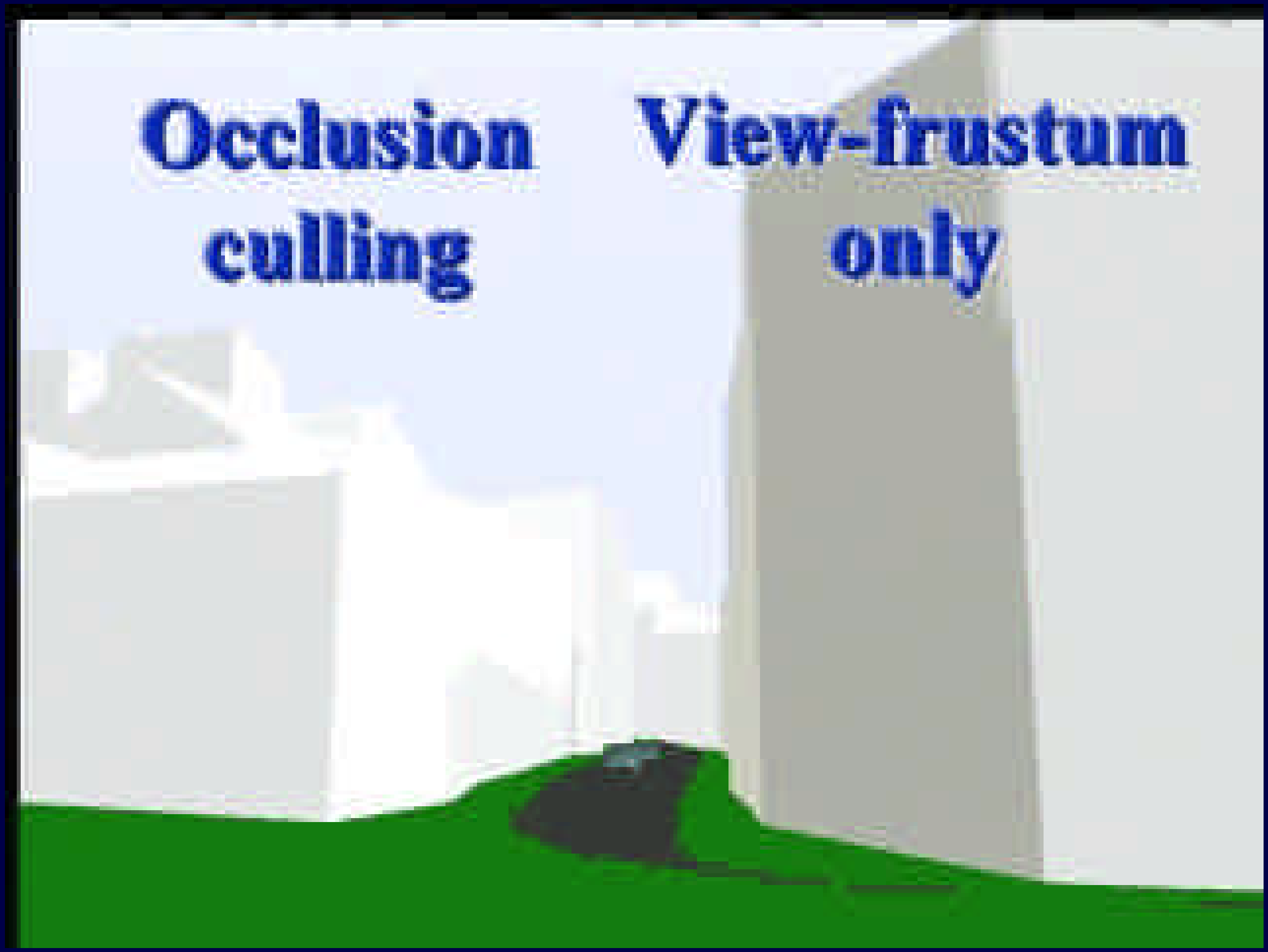


Results - Single projection plane

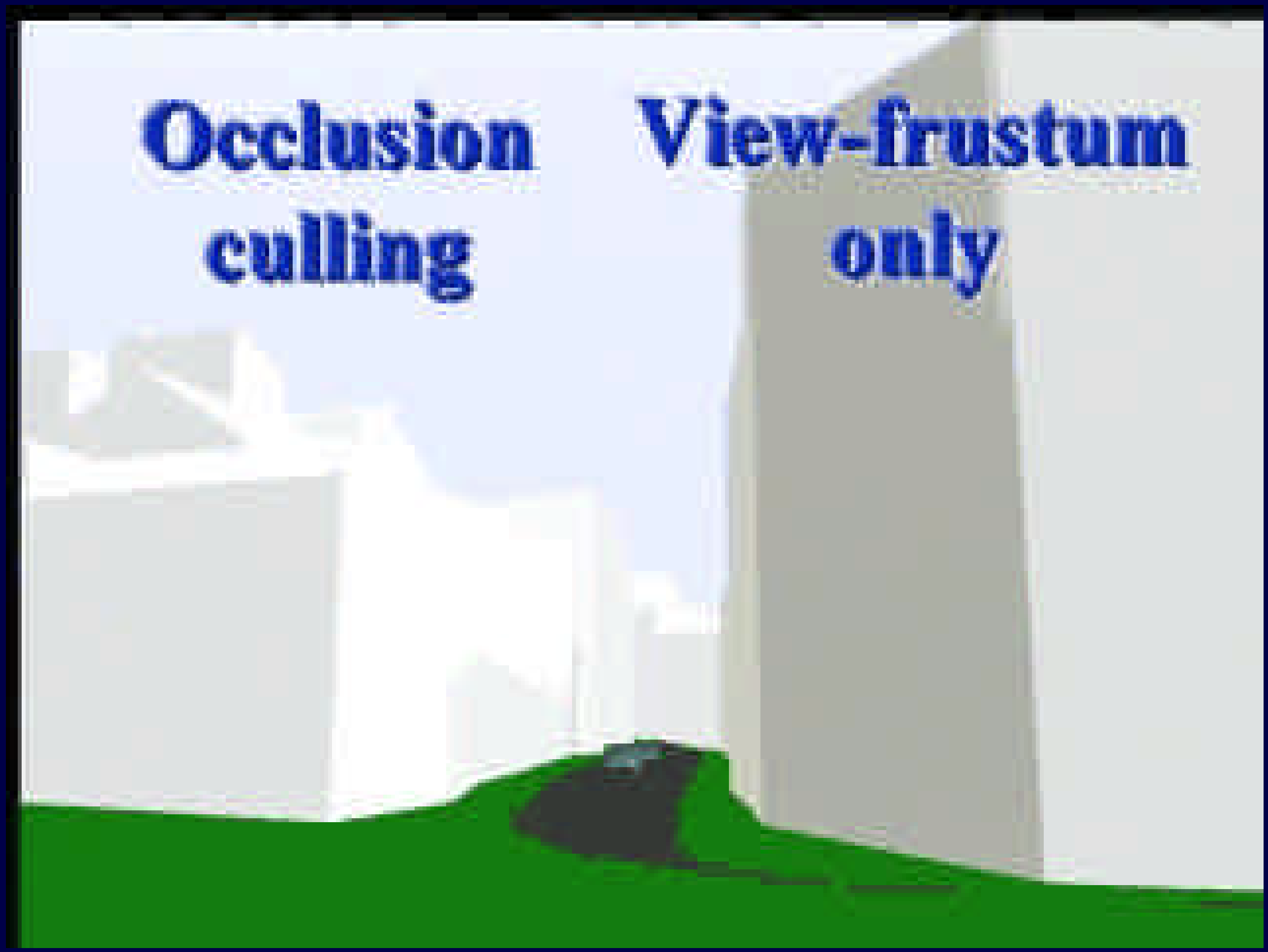
- City scene (6 million polygons)
- 165 minutes of preprocess (0.81 seconds per cell)
- 18 times speedup wrt view frustum culling
- Informal comparison with [Cohen-Or *et al.* 98] (no occluder fusion, single occluder):
  - 4 times fewer remaining objects
  - 150 times faster



Video



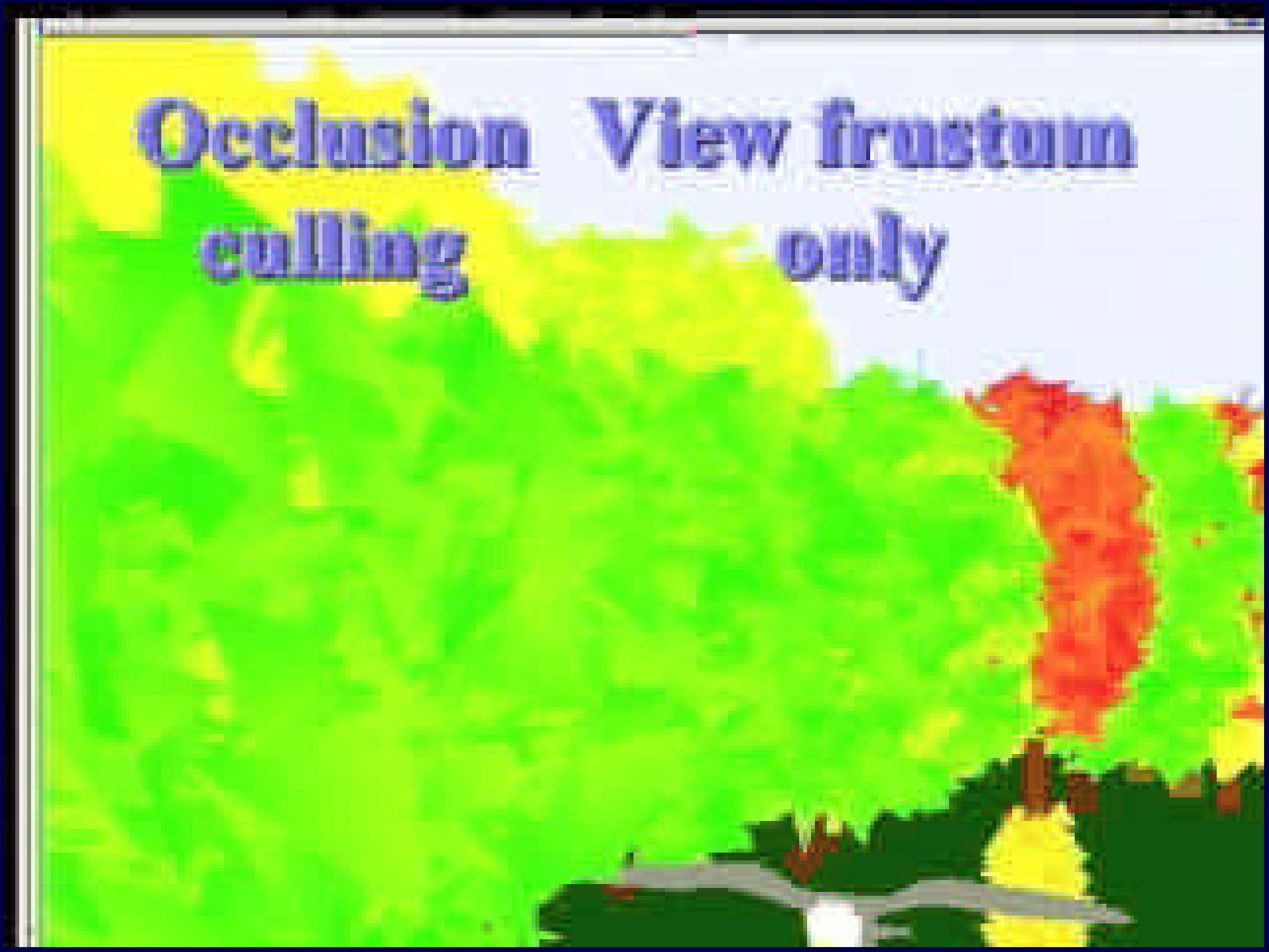
Video



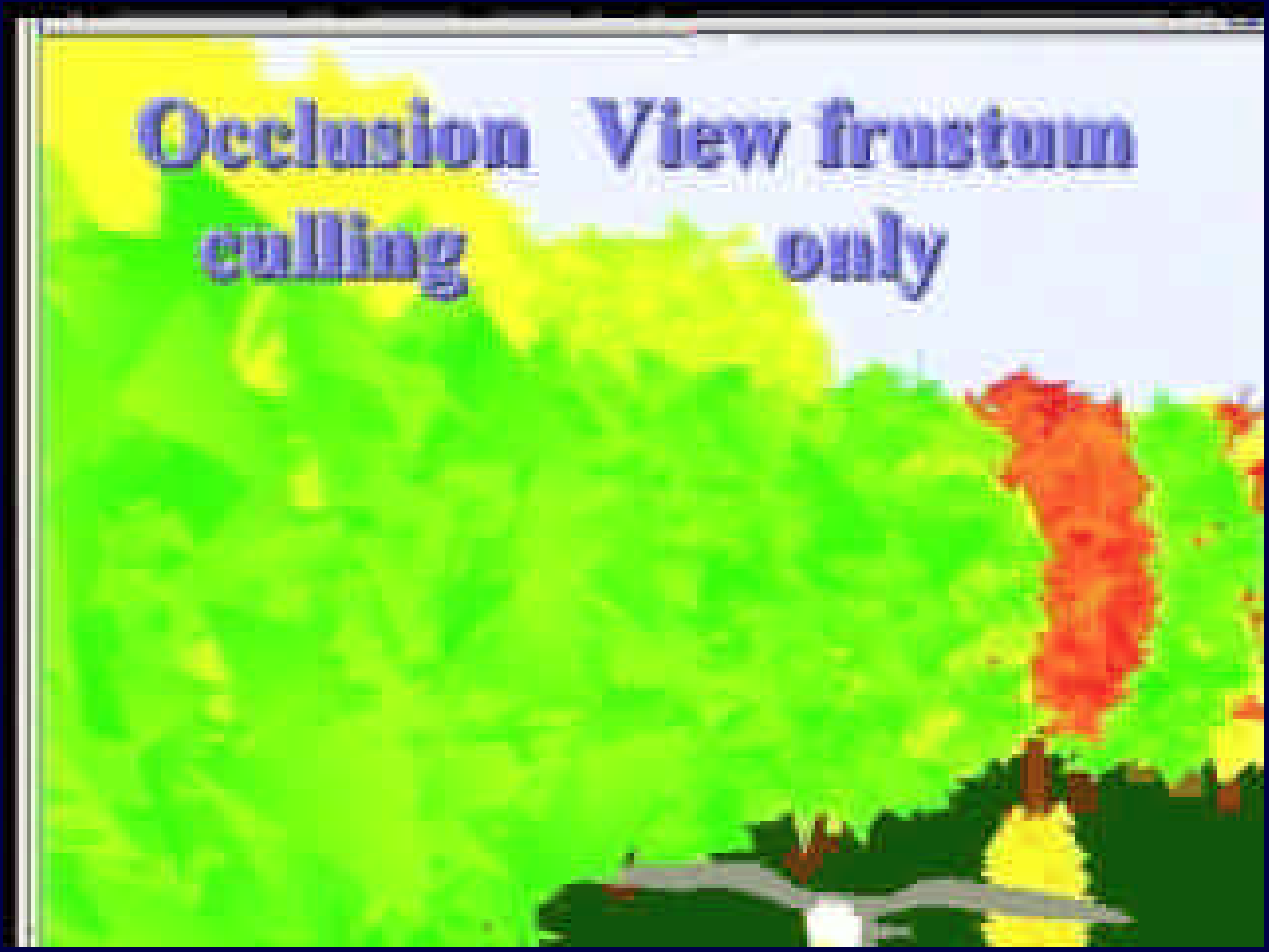
Results – Occlusion sweep

- Forest scene (7.8 million polygons)
- 15 plane positions
- 23 seconds per cell
- 24 times speedup wrt view frustum culling

Video



Video



Discussion

- \- More remaining objects than on-line methods
- \- No moving occluders

- \+ Occluder fusion
- \+ No cost at display time
- \+ Prediction capability
  - scenes which do not fit into main memory
  - pre-fetching (network, disk)

Future work

- Better concave occluder Projection
  - e.g. adaptation of [Lim 1992]
- On-demand computation
- Application to global illumination
- Use with other acceleration methods
  - LOD or image-based acceleration
  - Driven by semi-quantitative visibility
  - Take perceptual masking into account